

# Evaluating the Performance of Machine Learning Classifier Algorithms for Software Estimation in Software Development Projects

Muhammad Adeel Mannan <sup>1</sup>, Rohail Qamar <sup>2\*</sup>, Iqbal Uddin Khan <sup>1</sup>, Afzal Hussain <sup>1</sup>  
Saad Ahmed <sup>3</sup>, Jahangir Khan <sup>1</sup>

<sup>1</sup>Department of Computing, Faculty of Engineering Sciences & Technology, Hamdard University, Karachi, Pakistan; <sup>2</sup>Department of Computer Science & Information Technology, NED University of Engineering & Technology, Karachi, Pakistan; <sup>3</sup>Department of Computer Science, IQRA University, Karachi, Pakistan

**Keywords:** Software estimation, Classifier Algorithm, IG

## Journal Info:

Submitted:

February 15, 2024

Accepted:

March 25, 2024

Published:

March 31, 2024

## Abstract

The major aim of this research is to rank the best performing features in order to classify the Software estimation dataset using SVM, Naïve Bayes, Random forest, Decision tree, and KNN classifiers and evaluate their accuracy. Two steps are involved in the classification process: first, the dataset with all attributes is analyzed; second, the information gain methodology is used to rank the attributes, and only the highly rated ones are used to generate the model of classification. Using several folds of cross-validation, we assess the accuracy rank of SVM, Naive Bayes, Decision tree, Random forest, and KNN classifiers. Our results show that these classifiers perform rather well on the dataset.

\*Correspondence author email address: [rohailqamar@cloud.neduet.edu.pk](mailto:rohailqamar@cloud.neduet.edu.pk)

DOI: [10.21015/vtse.v12i1.1770](https://doi.org/10.21015/vtse.v12i1.1770)

## 1 Introduction

A crucial step in data mining is knowledge discovery database, includes the process of transforming unprocessed data to processed data into useful information. The two types of jobs that fall under data mining are descriptive and predictive. Anticipating a target characteristic based on explanatory factors is the goal of predictive activities [1]. Classification is one of these predictive modeling jobs; it involves searching a model that clarifies and defines ideas, making it easier to forecast the class whose labels are unknown [2].

We use the software estimate dataset available in the PROMISE data repository for our analysis [3]. Classifying the dataset using ML classifiers is the main goal of this study. Using feature selection as a preprocessing step in machine learning helps to improve classifier performance by lowering dimensionality and separating important information. Examining how the information gain ranking approach affects classifier accuracy is the goal of this work. We used a cross validation technique (10-fold) to evaluate the classifiers' performance.



In this research; Section 2 explores the review of literature. Section 4 shows the findings of the experiment and the discussions that followed. Section 3 describes the approach used for the categorization analysis. The final portion offers conclusions and directions for more study.

## 2 Research Background

Numerous research on effort estimate using statistical and neural network methods have been published, according to a survey of the literature. Research on software estimate and defect prediction has also used decision trees and other data mining techniques. Burk et al. [4] compared the predicted accuracy of many statistical models and artificial neural networks across periods of five and ten years. Lundin et al. [5] projected software effort across 15-year intervals using logistic regression models and artificial neural networks. Pendharkar et al. [6] found trends in software effort by applying a variety of data mining approaches. Researchers have used the PROMISE dataset in the past and have obtained accuracy rates between 66% and 78%.

For example, in their software estimating recurring event studies, Michalski et al. [7] achieved a 66% classification accuracy. Clark and Niblett [8] used a 70/30 split of a dataset including information on 286 software items in order to forecast effort recurrence occurrences throughout a 5-year timeframe. They achieved accuracy ranging from 65% to 72% across several techniques. A 78% accuracy rate was found by Cestink et al. Zhang and Su [9] examined the ranking, or Area Under the Curve (AUC), of decision tree learning algorithms and Naïve Bayes.

## 3 Research Methodology

Numerous research on effort estimates using statistical and neural network methods have been published, according to a survey of the literature. Research on software estimate and defect prediction has also used decision trees and other data mining techniques. Because teams and organizations have historically struggled with effort estimating, it is important to consider it early in a software development project [20]. Burk et al. [4] compared the predicted accuracy

of many statistical models and artificial neural networks across periods of five and ten years. Lundin et al. [5] projected software effort across 15-year intervals using logistic regression models and artificial neural networks. Pendharkar et al. [6] found trends in software effort by applying a variety of data mining approaches. Machine learning provides an alternate method rather than depending just on algorithmic models. For estimate, methods including decision trees (DT), Bayesian networks, genetic algorithms, ANN, support vector machine, and case-based reasoning (CBR) are used [25]. Researchers have used the PROMISE dataset in the past and have obtained accuracy rates between 66

For example, in their software estimating recurring event studies, Michalski et al. [7] achieved a 66% classification accuracy. Clark and Niblett [8] used a 70/30 split of a dataset including information on 286 software items in order to forecast effort recurrence occurrences throughout a 5-year timeframe. They achieved accuracy ranging from 65% to 72% across several techniques. A 78% accuracy rate was found by Cestink et al. Zhang and Su [9] examined the ranking, or Area Under the Curve (AUC), of decision tree learning algorithms and Naïve Bayes. Different performance aspects are measured and illustrated using different metrics [21]. The accuracy of performance evaluations depends on the assessment measures that are used [22].

## 4 Implementation Results

The experiment carried out to examine the classifiers' performance using WEKA tool kit is presented in this part.

### 4.1 Description of Dataset

The dataset used in this work, which included 10,885 samples and 12 characteristics, was obtained from the PROMISE Software Engineering Repository. Eleven of these characteristics are choice variables, while the remaining one is the target variable. [3, 17, 18] has detailed information on the benchmark functions, including mathematical descriptions of them. Unlike other benchmarks, these benchmark functions were chosen because their optimum values were known,

**Table 1.** Attribute Details

Attribute	Description	Rank Value
Ev(g)	Essential Complexity	0.08
Lv(g)	Design Complexity	0.07
V	Volume	0.06
L	Program length	0.05
D	Difficulty	0.04
I	Intelligence	0.03
E	Effort	0.02
T	Time Estimator	0.01

**Figure 1.** Attribute Ranking

making them the most reliable for confirming model acceptance, as shown in table 1.

## 4.2 Attribute Classification

### 4.2.1 Method I

In this case, every attribute in the dataset is used to classify the data. Several cross-validation folds are used to assess the performance.

### 4.2.2 Method II

An information gain ranking filter is used to rank the characteristics in the dataset. The representative attribute's most illuminating qualities are chosen [19]. Figure 1 presents the ranking of the qualities. When creating the classifier model, consideration is given to the eight qualities that score the highest. The score depends upon the weighting criteria ref24.

## 4.3 Classification Results

This inquiry has used a variety of measures to evaluate the algorithms' performance. The following is an outline of them:

- Accurately Classified Instances: The proportion of cases when the actual value and the projected value coincide.
- Inaccurately Classified Instances: The proportion of cases when the actual value and the anticipated value diverge.
- The average size of each individual inaccuracy is known as the mean absolute error (MAE).
- Root Mean Squared Error (RMSE): Determines the error in several calculations.
- The overall absolute error adjusted by the predictor's error is known as the relative absolute error (RAE).
- The total squared error divided by the total squared error of the predictor yields the root relative squared error (RRSE).
- Rates of True Positive (TP) and False Positive (FP)
- F-Measure represents accuracy.
- Area Under Curve (AUC): Indicates how well the classifier can discriminate between two possible outcomes.

Correct classifications are represented by True Positives (TP) and True Negatives (TN). When a result is wrongly forecasted as positive when it is actually negative, it is called a False Positive (FP), and when it is wrongly predicted as negative when it is actually positive, it is called a False Negative (FN).

The findings of method I's evaluation of the SVM in relation to various performance criteria are shown in Table 2. It shows the different values of the 12 selected performance quality measures concerning 5 k-folds from fold-6 to fold-10.

The seventh cross-validation fold yields the maximum accuracy with this approach, and as the number of folds grows, the accuracy does not significantly change. Table 3 presents the findings from the assessment of the SVM using method II for performance measures. Where Information gain ranking filter is used to rank the 8 characteristics selected from

**Table 2.** Performance metrics of SVM using method I

Accurately classified instances	233	237	236	237	236
Un-accurately classified instances	53	49	50	49	50
Testing method					
	6 fold (%)	7 fold (%)	8 fold (%)	9 fold (%)	10 fold (%)
MSE	17	17	17	17	17
RMSE	30	29	29	29	30
RAE	47	49	48	48	48
RRSE	71	69	70	69	71
TP Rate	90	93	92	93	92
FP Rate	11	7	8	7	8
Precision	89	92	91	92	91
Recall	90	93	92	95	92
F-Measure	83	85	84	85	84
AUC	88	89	89	89	88
Accuracy	81	83	83	83	83
Error Rate	19	17	17	17	17

**Table 3.** Performance metrics of SVM using method II

Accurately classified instances	49	49	49	49	49
Un-accurately classified instances	16.61	16.6	16.56	16.61	16.55
Testing Method					
	6 fold (%)	7 fold (%)	8 fold (%)	9 fold (%)	10 fold (%)
MSE	17	17	17	17	17
RMSE	29	29	29	29	29
RAE	48	48	48	48	48
RRSE	70	70	69	69	69
TP Rate	93	93	93	93	93
FP Rate	7	7	7	7	7
Precision	92	92	92	92	92
Recall	93	93	93	93	93
F-Measure	85	85	85	85	85
AUC	89	89	89	89	89
Accuracy	83	83	83	83	83
Error Rate	17	17	17	17	17

the dataset. The representative attribute's most illuminating qualities are chosen [23].

Other metrics are essentially the same in method II as well because the number of successfully categorized instances stays constant regardless of the number of folds. The evaluation of KNN employing method I is shown in Table 4 in relation to various performance criteria. It shows the different values of the 12 selected performance quality measures with respect to 5 k-folds from fold-6 to fold-10.

The seventh cross-validation fold yields the maximum accuracy with this approach, and as the number of folds grows, the accuracy does not significantly change. Table 5 presents the findings from the KNN evaluation using method II for performance indicators. Where the Information gain ranking filter is used to rank the 8 characteristics selected from the dataset. The representative attribute's most illuminating qualities are chosen.

Because II maintains the same correctly categorized instances regardless of the number of folds,

**Table 4.** Performance metrics of KNN using method I

Accurately classified instances	233	237	236	237	236
Un-accurately classified instances	53	49	50	49	50
Testing method					
	6fold (%)	7fold (%)	8fold (%)	9fold (%)	10fold (%)
MSE	17	17	17	17	17
RMSE	30	29	29	29	30
RAE	47	49	48	48	48
RRSE	71	69	70	69	71
TP Rate	78	78	78	78	78
FP Rate	24	25	25	25	25
Precision	89	92	91	92	91
Recall	78	78	78	78	78
F-Measure	83	85	84	85	84
AUC	88	89	89	89	88
Accuracy	81	83	83	83	83
Error Rate	19	17	17	17	17

**Table 5.** Performance metrics of KNN using method II

Accurately classified instances	238	238	238	238	238
Un-accurately classified instances	48	48	48	48	48
Testing Method					
	6 fold (%)	7 fold (%)	8 fold (%)	9 fold (%)	10 fold (%)
MSE	17	17	17	17	17
RAE	48	48	48	48	48
RRSE	70	70	68	69	68
TP Rate	78	78	78	78	78
FP Rate	7	7	7	7	7
Precision	92	92	92	92	92
Recall	78	78	78	78	78
F-Measure	85	85	85	85	85
AUC	89	89	89	89	89
Accuracy	83	83	83	83	83
Error Rate	17	17	17	17	17

other metrics are likewise essentially unchanged. Table 6 displays the findings of the ID3 Decision Tree classifier assessment conducted using method I. It shows the different values of the 12 selected performance quality measures with respect to 5 k-folds from fold-6 to fold-10.

The accuracy of this approach varies little as the number of cross-validation folds grows, with the best accuracy being reached in the ninth fold. Table 7

**Table 6.** Performance metrics of Decision Tree using method I

Accurately classified instances	232	234	234	237	232
Un-accurately classified instances	54	52	52	49	54
Testing Method					
	6 fold (%)	7 fold (%)	8 fold (%)	9 fold (%)	10 fold (%)
MSE	17	18	19	18	17
RAE	52	52	51	52	51
RRSE	72	72	71	71	70
TP Rate	81	81	82	83	81
FP Rate	22	22	21	20	22
Precision	76	76	77	78	76
Recall	85	88	86	87	86
F-Measure	81	82	81	82	81
AUC	89	89	89	89	89
Accuracy	81	82	82	83	81
Error Rate	19	18	18	17	19

**Table 7.** Performance metrics of Decision Tree using method II

Accurately classified instances	232	229	230	229	229
Un-accurately classified instances	56	57	56	57	57
Testing method					
	6fold (%)	7fold (%)	8fold (%)	9fold (%)	10fold (%)
MSE	16	18	17	18	19
RMSE	28	30	29	30	29
RAE	51	52	51	52	52
RRSE	72	72	71	72	70
TP Rate	81	80	81	81	80
FP Rate	22	23	22	22	23
Precision	86	86	87	86	86
Recall	85	85	86	85	85
F-measure	83	83	84	83	83
AUC	88	89	89	89	89
Accuracy	81	80	81	81	80
Error rate	19	20	19	19	20

**Table 8.** Performance metrics of Naïve Bayes using method I

Accurately classified instances	232	234	234	237	232
Un-accurately classified instances	54	52	52	49	54
Testing Method					
	6fold (%)	7fold (%)	8fold (%)	9fold (%)	10fold (%)
MSE	17	18	16	18	19
RMSE	29	30	28	30	28
RAE	52	52	51	52	51
RRSE	72	72	70	71	72
TP Rate	81	81	82	83	81
FP Rate	22	22	21	20	22
Precision	76	76	77	78	76
Recall	85	88	86	87	86
F-Measure	81	82	81	82	81
AUC	89	89	89	89	89
Error Rate	19	18	18	17	19

presents the findings from the ID3 Decision Tree Classifier assessment conducted with method II. Where the Information gain ranking filter is used to rank the 8 characteristics selected from the dataset. The representative attribute's most illuminating qualities are chosen.

With this approach, the fifth fold yields the lowest accuracy and the eighth fold yields the best accuracy. Table 8 displays the findings of the Naïve Bayes classifier assessment conducted using method I. It shows the different values of the 12 selected performance quality measures with respect to 5 k-folds from fold-6 to fold-10.

The accuracy of this approach varies little as the number of cross-validation folds grows, with the best accuracy being reached in the ninth fold. Table 9 presents the findings from the Naïve Bayes Classifier assessment conducted with method II. Where, Information gain ranking filter is used to rank the 8

**Table 9.** Performance metrics of Naïve Bayes using method II

Accurately classified instances	232	229	230	229	229
Un-accurately classified instances	56	57	56	57	57
Testing method					
	6fold (%)	7fold (%)	8fold (%)	9fold (%)	10fold (%)
MSE	17	18	19	18	17
RMSE	29	29	29	29	29
RAE	52	52	51	52	51
RRSE	72	72	71	72	71
TP Rate	81	80	81	81	80
FP Rate	22	23	22	22	23
Precision	86	86	87	86	86
Recall	85	85	86	85	85
F-measure	83	83	84	83	83
AUC	88	89	89	89	89
Accuracy	81	80	81	81	80
Error rate	19	20	19	19	20

**Table 10.** Performance metrics of Random Forest using method I

Accurately classified instances	232	234	234	237	232
Un-accurately classified instances	54	52	52	49	54
Testing Method					
	6fold (%)	7fold (%)	8fold (%)	9fold (%)	10fold (%)
MSE	16	18	16	18	17
RMSE	29	30	28	30	29
RAE	52	52	51	52	51
RRSE	72	72	72	71	71
TP Rate	81	81	82	83	81
FP Rate	22	22	21	20	22
Precision	76	76	77	78	76
Recall	85	88	86	87	86
F-Measure	81	82	81	82	81
AUC	89	89	89	89	89
Accuracy	81	82	82	<b>83</b>	81
Error Rate	19	18	18	17	19

characteristics selected from the dataset. The representative attribute's most illuminating qualities are chosen.

With this approach, the fifth fold yields the lowest accuracy and the eighth fold yields the best accuracy. Table 10 displays the outcomes of the Random Forest classifier assessment conducted using method I. It shows the different values of the 12 selected performance quality measures with respect to 5 k-folds from fold-6 to fold-10.

With this approach, the ninth cross-validation fold yields the maximum accuracy, and the accuracy does not significantly change as the number of folds rises. Table 11 presents the findings from the Random Forest Classifier assessment using method II. Where, Information gain ranking filter is used to rank the 8 characteristics selected from the dataset. The representative attribute's most illuminating qualities are chosen.

**Table 11.** Performance metrics of Random Forest using method II

Accurately classified instances	232	229	230	229	229
Un-accurately classified instances	56	57	56	57	57
Testing method					
	6fold (%)	7fold (%)	8fold (%)	9fold (%)	10fold (%)
RMSE	29	30	28	30	30
RAE	50	52	50	52	51
RRSE	72	72	71	72	71
TP Rate	81	80	81	81	80
FP Rate	22	23	22	22	23
Precision	86	86	87	86	86
Recall	85	85	86	85	85
F-measure	83	83	84	83	83
AUC	88	89	89	89	89
Accuracy	81	80	81	81	80
Error rate	19	20	19	19	20

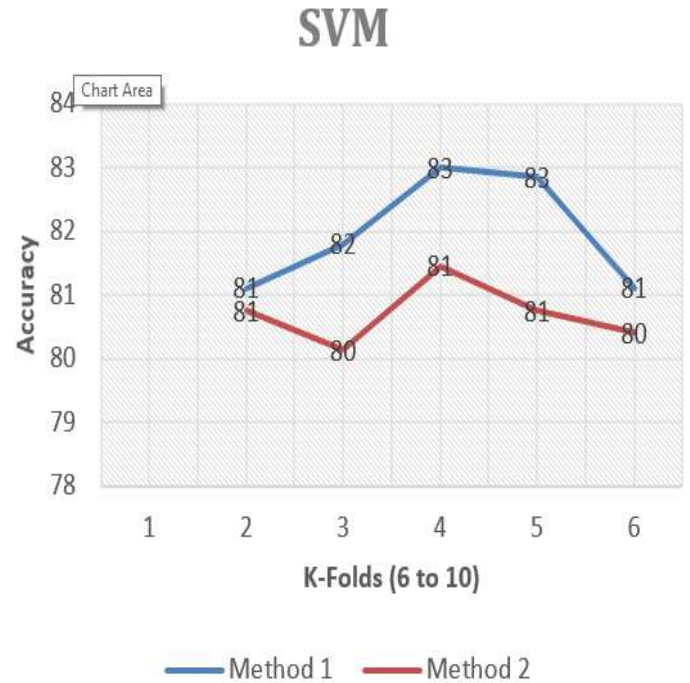
With this approach, the fifth fold yields the lowest accuracy and the eighth fold yields the best accuracy.

### 4.4 Result Discussion

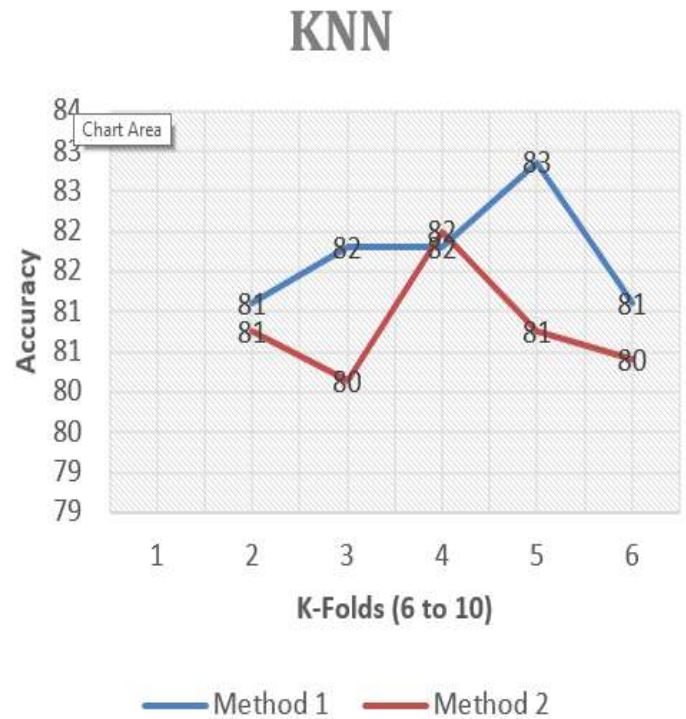
The results generated from the dataset using WEKA is shown graphically in the figures mentioned below. Here the 1st value in the horizontal axis shows the 5th fold and the 6th value shows the 10th fold. Figures 2 and 3 demonstrate that the accuracy of the SVM classifier is optimal in the 8th fold of cross validation, while the accuracy of the KNN classifier changes for different 9th folds of cross validation. It means that SVM classifiers achieve the accuracy results earlier as compare to the KNN classifier, that leads to reduce the software effort estimation. Figure 4 demonstrate that the accuracy of the ID3 decision tree classifier is optimum in the 7th fold of cross validation, which means the decision tree results are better than the previous two methods for the software effort estimation. Figures 5 and 6 demonstrate that the accuracy of the NB and random forest classifier is optimal in the 9th fold. It means they are less important as compared to all other algorithms.

### 5 Conclusion

We investigate the use of classifier algorithms in data mining approaches for categorizing a software estimation dataset. The SVM technique, using all the attributes in the dataset, shows an accuracy that varies between 81.4% and 82.56% in different folds; the maximum accuracy is found in the 5 and 7 folds. On the other hand, the accuracy rates of the NB algorithm vary from 79.37% to 82.86% in different



**Figure 2.** Accuracy Comparison using 2 methods



**Figure 3.** Accuracy Comparison using 2 methods

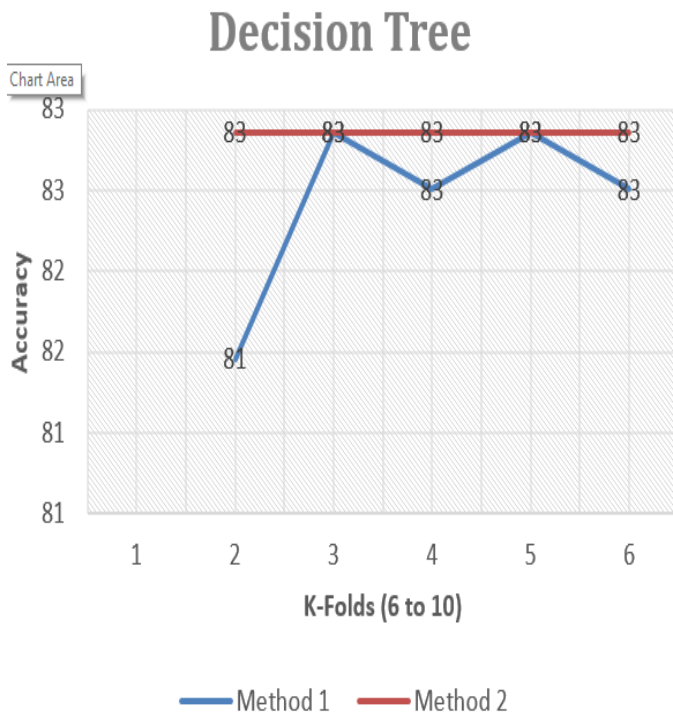


Figure 4. Accuracy Comparison using 2 methods

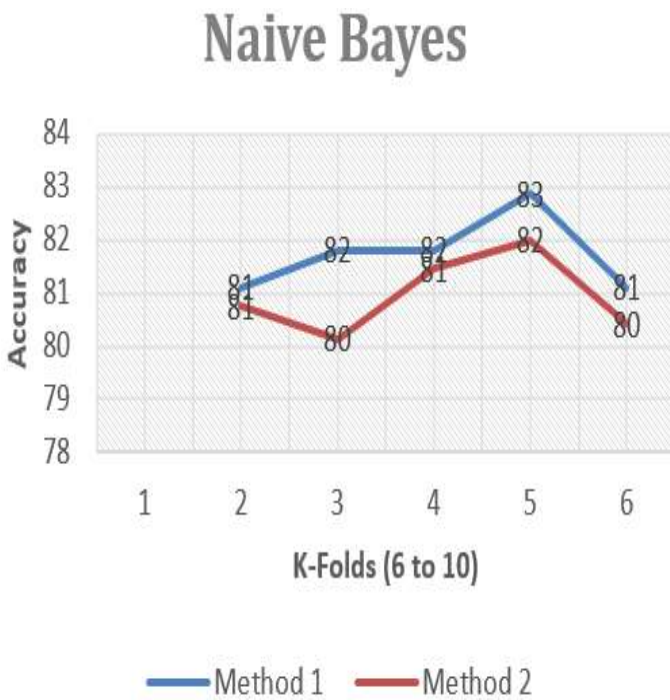


Figure 5. Accuracy Comparison using 2 methods

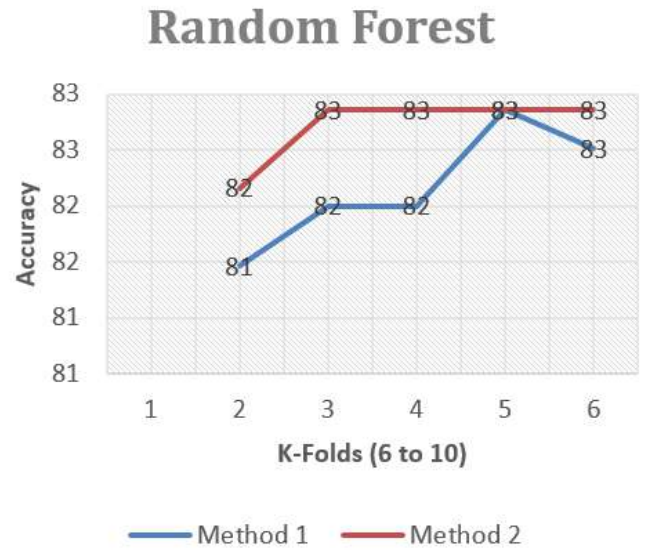


Figure 6. Accuracy Comparison using 2 methods

cross-validation folds; the ninth fold shows the highest accuracy.

No matter how many cross-validation folds are used, the ID3 Decision Tree method retains a constant accuracy rate when it uses just the characteristics that are ranked highest for classification. On the other hand, the KNN algorithm’s accuracy varies from 77.97% to 81.46%, with the eighth fold achieving the maximum accuracy. When evaluating characteristics with substantial information gain for classification, ID3 outperforms other classifiers in terms of accuracy within a few cross-validation folds, although Random Forest classifiers’ performance improves less noticeably.

Moreover, for all cross-validation folds, the accuracy of NB classifiers with attribute ranking declines. Future research might concentrate to improve the accuracy of NB and Random Forest classifiers. Furthermore, assessing the algorithms according to runtime.

### Author Contributions

**Muhammad Adeel Mannan:** Conceptualization, Methodology, Software. **Rohail Qamar:** Supervision, Data curation, Writing Original draft preparation. **Iqbal Uddin Khan:** Visualization, Investigation. **Afzal**

**Hussain:** Software, Validation. **Saad Ahmed:** Writing Reviewing and Editing. **Jahangir Khan:** Visualization, Investigation.

## Compliance with Ethical Standards

It is declared that all authors don't have any conflict of interest. It is also declared that this article does not contain any studies with human participants or animals performed by any of the authors. Furthermore, informed consent was obtained from all individual participants included in the study.

## Funding Information

Not applicable.

## References

- [1] J. Han and M. Kamber, *Data Mining Concepts and Techniques*, Morgan Kaufmann, 2001.
- [2] X. Xiong et al., "Analysis of software estimation using Data Mining & Statistical Techniques," in *IEEE Proceedings of 6th International Conference on Software Engineering*, 2015, pp. 82-87.
- [3] J. Sayyad Shirabad and T. J. Menzies, "The PROMISE Repository of Software Engineering Databases," School of Information Technology and Engineering, University of Ottawa, Canada. [Online]. Available: <http://promise.site.uottawa.ca/SERepository>
- [4] M. Prasad, "Online Feature Selection for Classification," *International Journal of Computational Intelligence Systems*, vol. 1, no. 2, 2018, pp. 127-133.
- [5] D. Delen, G. Walker, and A. Kadam, "Predicting software effort estimation: A comparison of three data mining methods," *Artificial Intelligence in Medicine*, vol. 34, no. 2, 2015, pp. 113-127.
- [6] H. B. Burk et al., "Artificial Neural Networks Improve the Accuracy of software Prediction," 79(4), 2019, pp. 857-862.
- [7] M. Lundin et al., "Artificial Neural Networks Applied to estimation of software efforts," *Sustainability*, vol. 57, no. 4, 2020, pp. 281-286.
- [8] P. C. Pendharkar et al., "Association, Statistical, Mathematical and Neural Approaches for Mining," *Expert systems with Applications*, vol. 17, 2019, pp. 223-232.
- [9] P. Clark and T. Niblett, "Induction in Noisy Domains," in *Progress in Machine Learning*, eds. I. Bratko & N. Lavac, Sigma Press, 2017, pp. 11-30.
- [10] P. Clark and T. Niblett, "The CN2 induction algorithm," *Machine Learning Journal*, vol. 3, no. 4, 2018, pp. 261-283.
- [11] G. Cestnik et al., "Assistant-86: A Knowledge Elicitation Tool for Sophisticated Users," in *Progress in Machine Learning*, eds. I. Bratko & N. Lavrac, Sigma Press, 2017, pp. 31-45.
- [12] H. Zhang and J. Su, "Naïve Bayesian Classifiers for Ranking," in *Proceedings of 15th European Conference on Machine Learning*, Springer, 2014, pp. 501-512.
- [13] L. Jianq and Y. Guo, "Learning Lazy Naïve Bayesian Classifiers for Ranking," in *Proceedings of 17th International Conference on Tools with Artificial Intelligence*, 2015, pp. 412-416.
- [14] J. Huang et al., "Comparing Naïve Bayes, Decision Trees and SVM with AUC and Accuracy," in *Proceedings of 3rd International Conference on Datamining*, IEEE Computer Society Press, 2013, pp. 553-556.
- [15] D. T. Larase, *Discovering Knowledge in Data. An introduction to Data mining*, John Wiley & Sons, Inc, 2015.
- [16] I. H. Witten and E. Frank, *Datamining: Practical Machine Learning Tools and Techniques*, 2nd edn., Elsevier, 2015.
- [17] Mannan, A., Qamar, R., & Arshad, S. (2024). Semi-Automated Approach for Evaluation of Software Defect Management Process using ML Approach. *VAVKUM Transactions on Computer Sciences*, 12(1), 20-33.
- [18] R. Qamar, R. Asif, L. F. Naz, A. Mannan, & A. Hussain, "FlightForecast: A Comparative Analysis of Stack LSTM and Vanilla LSTM Models for Flight Prediction," *VFAST Transactions on Software Engineering*, vol. 12, no. 1, pp. 13-24, 2024.
- [19] M. A. Mannan and A. Ansari, "SPMM: A Model Taxonomy for Designing and Managing Quality System," in *IEEE Access*, vol. 10, pp. 76720-76730, 2022. doi: 10.1109/ACCESS.2022.3190081.
- [20] E. C. Ltd, "Software effort estimation," *GitHub*, Available: <https://github.com/edusoftresearch/SEEData>, 2023.

- [21] Y. Mahmood, N. Kama, A. Azmi, A. S. Khan, and M. Ali, "Software effort estimation accuracy prediction of machine learning techniques: A systematic performance evaluation," *Software: Practice and Experience*, vol. 52, no. 1, pp. 39–65, 2022.
- [22] A. Jadhav, M. Kaur, and F. Akter, "Evolution of software development effort and cost estimation techniques: five decades study using automated text mining approach," *Mathematical Problems in Engineering*, vol. 2022, pp. 1–17, 2022.
- [23] P. Suresh Kumar, H. Behera, J. Nayak, and B. Naik, "A pragmatic ensemble learning approach for effective software effort estimation," *Innovations in Systems and Software Engineering*, vol. 18, no. 2, pp. 283–299, 2022.
- [24] P. V. AG and V. Varadarajan, "Estimating software development efforts using a random forest-based stacked ensemble approach," *Electronics*, vol. 10, no. 10, p. 1195, 2021.
- [25] Z. R. Mohsin, "Comparative study for software effort estimation by soft computing models," *Journal of Education for Pure Science-University of Thi-Qar*, vol. 11, no. 2, pp. 108–120, 2021.