Journal of Applied Data Sciences
Vol. 2, No. 4, December 2021, pp. 157-173

An Ensemble and Filtering-Based System for Predicting Educational Data Mining

Andhika Rafi Hananto [1,*], Silvia Anggun Rahayu [2], Taqwa Hariguna [3]

Department of Information Systems, Universitas Amikom Purwokerto, Indonesia
[1] andhikarh90@gmail.com *; [2] silviaanggunr@gmail.com; [3] thariguna@gmail.com
* corresponding author

(Received: August 30, 2021; Revised: September 16, 2021; Accepted: October 20, 2021; Available online: December 31, 2021)

Abstract

When developing a prediction paradigm, an ensemble technique such as boosting is used. It is built on a heuristic framework. Generally speaking, engineering ensemble learning is more accurate than individual classifiers when it comes to making predictions. Consequently, numerous ensemble strategies have been presented in this work, particularly to provide a more complete understanding of the essential methods in general. Researchers have experimented with boosting methods to forecast student performance as part of a variety of ensemble techniques. The researchers employed improvement approaches to construct an accurate predictive educational model, which was based on a key phenomena seen in categorization and prediction operations. In light of the uniqueness and originality of the suggested strategy in educational data mining, the researchers used augmentation strategies in order to construct an accurate predictive pedagogical model. Tenfold cross-validation was performed to evaluate the effectiveness of the basic classifiers, which included the random tree, the j48, the knn, and the Naive Bayes. The random tree was found to be the most effective classifier. Several additional screening techniques, including oversampling (SMOTE) and undersampling (Spread subsampling), were utilized to analyze any statistically significant variations in results between the meta and base classifiers that had been identified between the meta and base classifiers. The use of ensemble and screening strategies, as compared to the use of standard classifiers, has demonstrated considerable gains in predicting student performance, as has the use of either strategy alone. Furthermore, after the completion of a performance research on each approach, two new prediction models have been established on the basis of the improved results gained thus far.

Keywords: Education Data Mining; J48, Naive Bayes; Random Tree; K-NN; Ensemble; Boosting.

1. Introduction

With the development of pedagogical data, the need to find productive information has emerged as a hot topic among the research community [1]. Many data mining techniques have been exploited in this direction to achieve better insight of different academic data warehouses [2,3]. Extraction of significant knowledge from the warehouse plays a major role in propelling the wheel of further education by using various data mining techniques [4,5].When it comes to educational data mining, it is generally agreed that predicting student achievement is a critical responsibility. Researchers Zhu et al. [5], conducted a case study to predict drop-out ratios using various classification approaches [6]. It was determined that the decision tree algorithm was correct and produced significantly different results with Bayes net and JRip rules being two types of rules. The CRISP methodology was exploited to test student performance in a c++ course [7], and the researcher in this study compared the performance of two decision trees viz. ID3 and C4.5 with naive bayes algorithm. The researchers Zhu et al. [5], also explored analog classifiers including ID3 and C4.5 on datasets relating to MCA students, with the primary objective of predicting and improving student performance [8]. Kotsiantis et al [7], used the Naive Bayes ensemble method, neural network and the WINDOWS algorithm to predict student performance [9].

In another research study, the authors proposed a prediction system based on the Adaboost algorithm to reduce risk failure by providing timely advice to high-risk students [10]. Additional study was done on educational data to increase prediction accuracy using different ensemble methodologies [11-13]. The authors also recommended several prediction paradigms based on their findings. In order to forecast student performance, classification, grouping, and regression are the approaches that are most often utilized. The power of the ensemble technique, on the other hand, has only been employed by a few academics to forecast student performance in the past. The ensemble technique is advantageous because it gives the much-needed incentive for improving the prediction accuracy of several classifiers after the aggregated outputs have been created, and this is because Although the data is complicated, ensembles have the capacity to achieve higher accuracy and, as a result, to avoid overfitting of individual classifiers and generalization mistakes in the classification process. Several research papers have proven that employing the ensemble strategy to combine the outputs of separate classifiers resulted in a decrease in generalization error [14-16].

One of the most important factors in ensuring that the ensemble mechanism is used effectively is that there must be a wide range of classifiers used in the ensemble. For example, data diversity may be performed by providing the classifier with diverse subsamples of the input data, similar to how boosting is done in this instance. Essentially, there are a number of significant and basic differences that distinguish the various types of ensemble techniques from one another. When the outputs of various classifiers are combined to form a prospective prediction, this is referred to as an imperative factor. Another consideration is the amount of classifiers that need to be synthesized. Furthermore, the classifiers that are part of the The members of an ensemble may operate in a sequential or concurrent manner. Ensembles can also be divided into two types based on the learning process that was used during the training and testing phases of a subsample of the dataset base; these are referred to as meta ensemble learning and subsample learning, respectively, and are referred to as meta ensemble learning and subsample learning, respectively.

1.1. Ensemble Learning at Its Most Fundamental

This system's fundamental training and testing of classifiers results in the merging of numerous classifiers' outputs into a single classifier with improved scenario prediction accuracy. The suggested model in figure 1 fully explains the training and testing of classifiers during the early phases of basic ensemble learning. Figure 1, The suggested model incorporates a number of ensemble approaches, which are discussed in further detail in the next part.

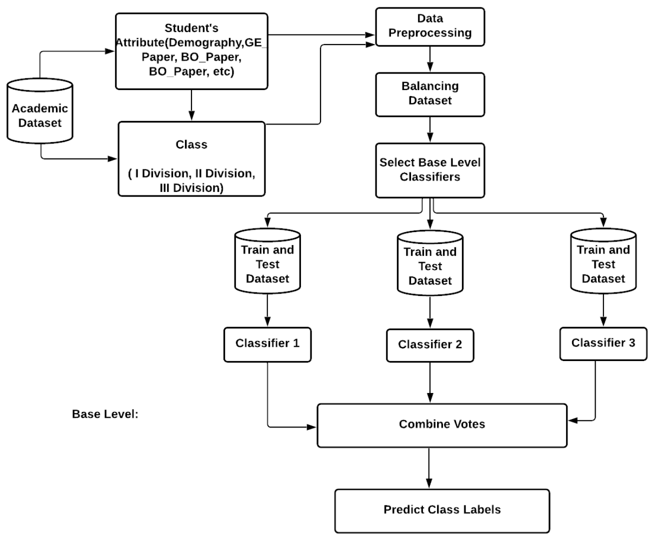

Figure. 1. Demonstrates foundation ensemble learning

1.1.1. Voting

Among the fundamental techniques for distributed skew algorithms is the voting approach, which is used to determine the average class allocation of the whole learning algorithm as a whole. Instead of allocating their votes to a single class that has been determined to be the most probable, the votes are distributed among a number of basic classifiers in order to estimate a given instance's final class is defined as To be more specific, the voting method is the only one that does not use cross-validation throughout the process's training and instance testing stages. Furthermore, combining algorithms may be performed in two ways: either by synthesizing many classifiers or by using meta synthesizers, or a combination of both. When dealing with difficulties when each classifier does the same work and earns the same amount of money, the straight-forward coupling strategy is the most effective method of solving the problem. The approach, on the other hand, is susceptible to inconsistencies and outliers. Contrastingly, the meta synthesizer is claimed to be potentially stronger in terms of the difficulties related with the classifier, namely, a more accurate fit and an increased training time. Voting determines the average potential class allocation for occurrences that are not defined as being in the following categories:

$$\overline{Prediction} = \sum_{k=0}^{N} \frac{Pk}{N} \quad (1)$$

1.1.2. Bagging

Bagging is a widely used ensemble approach that analyzes subsamples at the same time in order to improve the accuracy of the classifiers that are normally formed by merging several models. When various classifiers are hybridized together, a large number of outputs are formed by voting among the participants. According to Du et al. [13], an uneven learning process is required in order to generate statistically meaningful findings [17]. The basic classifier is also trained repeatedly on additional samples of the original dataset utilizing random sampling with a replacement technique in this method, which is a significant improvement over previous methods. Before training the basic classifier, it is necessary to alter it according to the formula below in order for random sampling from the original training set to be recreated. Adjustment of the basic classifier training set

$$NewTrainingSet = \{(AttrVec_{qr}, Class_{qr}) | qr = Rand(0, n), \quad 0 \leq r < n\} \quad (2)$$

1.1.3. Boosting

As arcing, this technique was developed by Schapire in 1990 to increase the classification accuracy performance of a machine learning classifier. In order for it to operate, weak learners must be run repeatedly on some training data that has been distributed. Random sampling from the original training set should be recreated. Adjustment of the basic classifier training set of strong classifiers in order to obtain much higher accuracy than the model generated by the individual weak learners. After a nearly 5-year hiatus, Yang [15] developed an improved version of the boosting algorithm known as AdaBoost [18], which is currently in use worldwide. The underlying principle of this learning method is to assign a weight to each occurrence in the training set, which is a simple concept. All instances are assigned the same priority at the start of the process, but with each iteration, the weight associated with the incorrectly classified instance is increased, while the weight associated with the properly identified instance is decreased (see Figure 1). As a result, in order to obtain the highest possible prediction accuracy, weak learning algorithms are pushed to concentrate on incorrect instances from the training set. Aside from that, the categorization

ISSN 2723-6471

is carried out in accordance with a voting method for each student (Ci), which has an individual weight ofIn mathematics, the relationship between I and is described as follows:

$$H(x) = sign(\sum_{i=0}^{i} \alpha i . c_i(x) \quad (3)$$

1.1.4. X-Validation

In machine learning, an extended ensemble approach, also known as selection by cross-validation, is a procedure in which an optimal selection among basic classifiers is made utilizing an inner 10-fold cross-validation system inside the ensemble. In order to create equal-sized training sets, each fold is split into two halves, which are then divided again. Gams [16] primarily depended on this strategy, which they further improved, in order to generate neural network ensembles [19,20]. To make matters even better, Domingos [17] employed cross-validation to speed up the development of his proposed rule induction system, known as RISE [21]. In this study, the training sets were split into equal-sized divisions depending on the number of people that took part in it. This is followed by the algorithm being executed on each split on a different thread. m+1 partition instances were used to assess the advanced rule of m partitions, which was designed to decrease over-fitting challenges and, as a consequence, boost accuracy of the model by decreasing over-fitting difficulties.

1.2. Meta Ensemble Learning

Predictions are made with specificity Rather than categorizing instances at the base level, this technique classifies them at the meta level, and a single classifier, referred to as the meta classifier, is responsible for classifying all of the instances and then aggregating them using a voting procedure to get the predictions. Figure 2 depicts the suggested model in a way that accurately illustrates the whole approach that was followed, from data pre-processing through the formulation of classification label predictions. The use of identical practice in producing unknown class labels is a common feature of many ensemble strategies, and as a result, multiple ways have been examined in detail in the following subsections.

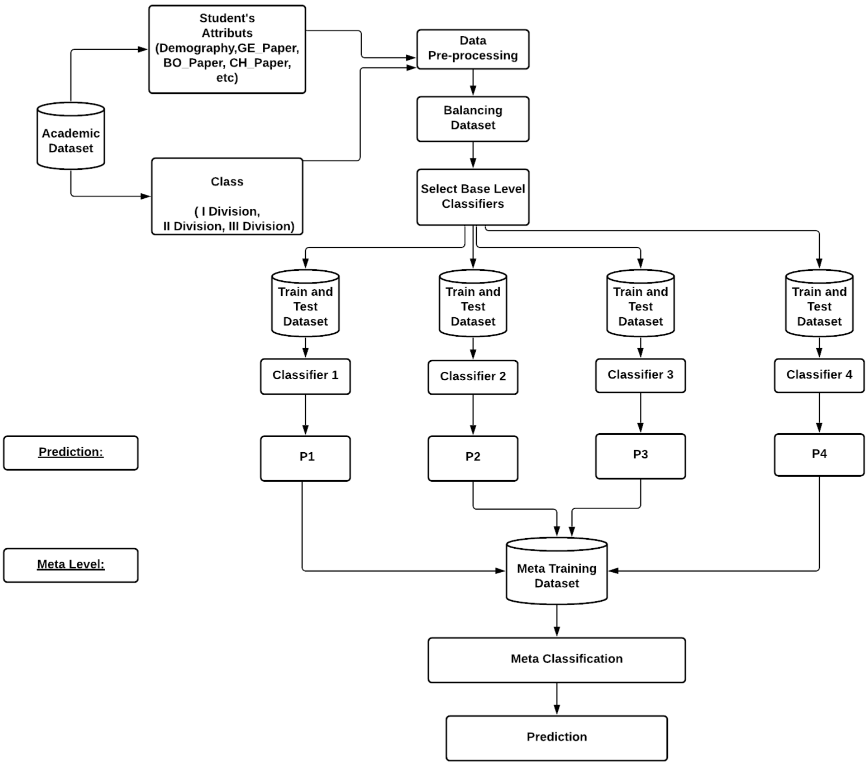

Figure. 2. Displays the suggested paradigm for Meta ensemble learning in action

1.2.1. Stacking

It is a procedure that is enforced with the goal of reaching the highest level of generalization accuracy possible. Zenko and Dzeroski [15] employed the stacking technique, in which they constructed an ensemble of learning algorithms, and ultimately determined in their analysis that observed stacking was the best classification methodology for predicting sample output [22]. In most cases, this strategy is used to hybridize the models formed by a range of classifiers and then attempt to establish if the classifiers' findings are consistent or inconsistent. This stage involves creating a meta training set using the predictions generated by each classifier on the original dataset. This meta training set will then be used to train the classifiers in the next step. When given training data, the meta classifier applies it to the classification issue in order to combine the multiple predictions and come up with a final decision. By producing probabilities for each class, which are often determined by separate base classifiers and may be used to increase performance, the stacking approach can usually be improved. Although the number of examples in the meta training set will increase as a result of such a situation, the number of instances in the training set will decrease.

1.2.2. Grading

The execution of a graded learning algorithm, according to Afsahhosseini and Al-Mulla [16], categorizes instances at the meta-level by using the meta classifier as a classification criteria, which is a classification criterion for the meta classifier [23]. The primary rationale behind the grading is that it teaches the meta classifier to categorize specific bases in order to forecast the instance whenever the base classifier fails in a particular job, which is the goal of the grading. A strategy based on weighted voting is used as the base classifier in this method. The base classifier is responsible for making the final classification prediction. In essence, a weighted vote is nothing more than the weight obtained by the When a correct prediction is produced, the confidence of the base classifier is calculated, which is generally computed by the related meta classifier when a successful prediction is made. Moreover, in his work, Yu et al. [17] proposes grading as a generalization method for cross-validation selection, in which the training set is divided into n subsets and an n-1 classifier is constructed, therefore removing one split at a time to investigate the speed of misclassification [24]. Finally, this strategy chooses the learning classifier with the lowest rate of misclassification. Grading only makes use of classifiers who possess a sufficient level of expertise in correctly categorizing a specific occurrence. Furthermore, for each base classifier, a variety of meta-data sets are created, which are then used to train meta-classifiers that are based on the metadata sets maintained by the base classifier.

1.2.3. Arbiter Trees

As previously mentioned, Du et al. [13] adopted this strategy, in which the arbiter tree was created in a bottom-up fashion [25]. For the most part, the dataset is arbitrarily truncated into many sub-partitions of size 'n'. Because of the combination of the two learning algorithms, the arbiter is stimulated, and subsequently recursively induced arbiters are produced from the output of numerous arbitrators, leading to an infinite number of arbiters being produced. There is a log2(n) degree of complexity in an arbitrator formed from n classifiers. It is necessary to pool training data and classifiers before building an arbitrator, which must then be classified by several classifiers once it has been constructed. The classification provided by the twin classifier is compared to the selection criteria, and a sample from the amalgamation pool is selected further to build a training set for the arbitrators based on the results. Every time the basic classifier produces a different classification, the arbitrator is responsible for compiling alternative classifications. As indicated in figure 3, the arbitrator, in conjunction with the arbitration rules , is responsible for determining the final classification result based on the basic predictions, which is shown in figure 3.

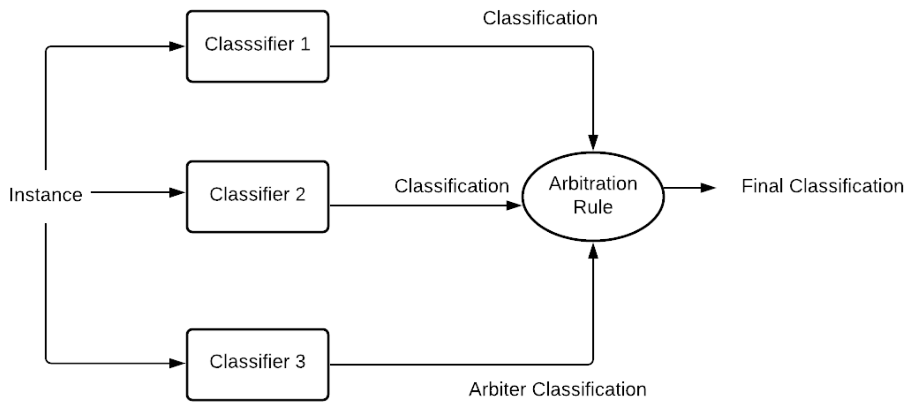

Figure. 3. Zhang et al [21] provides two basic learning algorithms as well as a single arbitrator

2. Methodology

In this thorough analysis, which evaluated critical elements linked with the dataset that are responsible for student performance in particular, a pedagogical dataset developed by the University of Kashmir was used as the basis. With great care and attention to detail, the researchers carried out this research investigation. Before being employed for analysis or observation, the whole dataset was subjected to a technique of pre-processing in order to exclude impotent features such as name, parentage, contact number, and so on from consideration. Additional tuples included missing data, which were removed in order to provide more accurate and dependable results in the future, According to the findings of the study, there were nine and twenty-four characteristics in the raw dataset before pre- and post-processing of the data, respectively, before pre- and post-processing of the data. Additionally, in the same way that demographic information about students was derived from registration numbers, English language skills and zoology, botany, and chemistry skills, among other variables related to academic dataset, were discovered by mining other attributes present in the raw dataset and then ranked based on their significance, as was the case with the variables mined from other attributes in the raw dataset. The multiple characteristics that have been obtained, as well as the possible values connected with each variable, are shown in the accompanying Table 1.

Table. 1. This function displays all of the potential values associated with each attribute of our dataset.

No.	Fields	Description
1	Demography	Rural, Urban
2	GE-Paper A	0-75
3	GE-Paper B	0-75
4	GE-Total	0-150
5	BO-Paper A	0-50
6	BO-Paper B	0-50
7	BO-Practical	0-25
8	BO-Intern	0-25
9	BO-Total	0-150
10	ZO-Paper A	0-50
11	ZO-Paper B	0-50
12	ZO-Intern	0-33
13	ZO-Practical	0-25
14	ZO-Total	0-150
15	CH-Paper A	0-150
16	CH-Paper B	0-50
17	CH-Paper C	0-50
18	CH-Intern	0-25
19	CH-Practical	0-25
20	CH-Total	0-150
21	Total-Marks	600

22	Total-Obtain	0-600
23	Overall Grade	Division I, II, III

3. Result & Discussion

3.1. The performance of learning classifiers on an individual basis

For predicting student performance, academics have employed a range of base and meta learning approaches, including the j48 algorithm, the random tree, and the logistic regression model. The course covers techniques such as naive bayes, K-nearest neighbor, and boosting, as well as machine learning methods. It was achieved via the use of a variety of filtering approaches, including the synthetic minority oversampling technique (SMOTE) and the Spread Subsampling method (SST) (under-sampling technique). In order to generate statistically meaningful findings, the oversampling and undersampling approaches were used to a genuine academic dataset obtained from Kaggle and based on the University paper test dataset. Furthermore, if the dataset is unbalanced before training and testing, the findings obtained might be skewed in favor of the majority class, increasing the likelihood of making an incorrect prediction. Taking these considerations into account, the two filtering procedures described above were implemented.

Table. 2. Results of multiple learning classifiers

CN	CC	IC	TPR	FPR	Precision	Recall	F-Measure	ROC	RAE
J48	93.21%	7.78%	0.93	0.06	0.93	0.92	0.93	0.95	14.52%
Naive Bayes	96.51%	5.44%	0.96	0.04	0.96	0.96	0.96	0.99	8.93%
Random Tree	91.31%	10.68%	0.91	0.07	0.91	0.91	0.91	0.93	16.45%
KNN	92.81%	9.17%	0.92	0.06	0.92	0.92	0.92	0.94	14.18%

Results of multiple learning classifiers are provided in the previously stated Table 2 after various learning algorithms have been executed across a pedagogical dataset. According to the results of Table 1, the naive bayes approach has obtained an amazing prediction accuracy of 95.50% when it comes to categorizing the relevant conditions. In terms of erroneous classification error and relative error, the classifier exceeds the competition, with 5.44% and 7.78%, respectively, having the lowest values among the other learning classifiers. As well as accuracy and recall, as well as the f-measure and the receiver operating characteristic (ROC) curves, a range of additional metrics related to learning classifiers have shown statistically significant results. These measurements include the Tp rate and the Fp rate, among others. The random tree achieved a noteworthy prediction accuracy of 91.31%, as well as a classification error of 10.68% and a relative absolute error of 15.46%, both of which were impressive. It also achieved a classification error of 10.68% and a relative absolute error of 15.46%. Other estimations associated with the classifier were also complimented for their accuracy; nevertheless, the results acquired were found to be less significant than those obtained by other learning classifiers, which was a source of disappointment for the researchers.

3.1.1. The results of basic classifiers created using the SMOTE approach are shown below.

The performance of the various classifiers after they have been exposed to one of the screening procedures, namely SMOTE, is described in Table 3 below across academic datasets. Following the implementation of the oversampling approach, the Precision of learning classifiers' predictions is increasing, which is also reflected by the improvement of other performance-related matrices This is shown by the data presented in Table 2. Several learning classifiers, including j48 (which improved from 93.21% to 93.99%), random tree (which improved from 91.31% to 91.85%), naive bayes (which improved from 96.51% to 98.16%), and Knn (which improved from 92.20% to 92.98%), demonstrated an improvement in performance (92.81% to 93.80%). Similarly, the relative absolute error associated with the basic classifier shows the lowest error after the application of SMOTE, decreasing from 14.52% to 11.53% (j48), from 16.45% to 14.02% (random tree), from 8.93% to 4.70% (naive bayes), and from 14.18% to 11.08% after the application of SMOTE, respectively (knn). The area under the curve (AUC) of the naive bayes classifier differs by a little margin, ranging from 0.99 to 0.97 in the case of the naive bayes classifier, according to ROC analysis (ROC).

Table. 3. Results obtained using the SMOTE approach are shown

CN	CC	IC	TPR	FPR	Precision	Recall	F-Measure	ROC	RAE
J48	93.99%	7.01%	0.93	0.03	0.92	0.92	0.93	0.95	11.53%
Naive Bayes	98.16%	2.84%	0.97	0.01	0.97	0.97	0.97	0.97	4.70%
Random tree	91.85%	9.15%	0.90	0.04	0.90	0.90	0.90	0.93	14.02%
KNN	93.80%	7.20%	0.92	0.03	0.92	0.92	0.92	0.94	11.08%

3.1.2. The results of the basic classifier using the Spread Subsampling approach are shown below.

For the purpose of determining whether or not there was an extra improvement in prediction accuracy in this section, the Spread Subsampling technique, which is an under-sampling procedure, was used in an educational dataset throughout this section. The following table contains estimates of the performance of a number of different classifiers. Taking a look at Tables 4 and 2, it is clear that post spread sub sampling increased the prediction accuracy of the knn classifier by an impressive margin (from 92.81% to 94.95%), demonstrating superiority over the knn classifier (over-sampling approach), which increased its accuracy by an additional margin (from 92.81% to 94.00%). A number of additional performance metrics linked with the knn classifier, such as the Tp rate, the Fp rate and the accuracy, among others, have also shown outstanding results. There are references to the same information in both Table 4 (Post Spread subsampling) and Table 2 (Pre Spread subsampling) (before Spread subsampling). Those who use the undersampling strategy, on the other hand, have reported inconsistent results among random trees, with their performance dropping from 91.31% to 89.96% as a consequence of the process. This despite the fact that the prediction accuracy of learning classifiers such as the j48 and naive bayes has shown considerable increases, growing from 93.21% to 93.68% and from 96.51% to 96.86% respectively, since it was first measured at the start of the research. Nonetheless, the results are not statistically significant in the same way as the findings obtained by the oversampling technique were.

Table. 4. The results of the Spread Sub-sampling approach are shown

CN	CC	IC	TPR	FPR	Precision	Recall	F-Measure	ROC	RAE
J48	93.68%	7.32%	0.92	0.03	0.92	0.92	0.92	0.95	11.78%
Naive Bayes	96.86%	4.14%	0.95	0.02	0.96	0.95	0.95	0.99	6.11%
Random tree	89.96%	11.04%	0.89	0.05	0.88	0.88	0.89	0.91	16.75%
KNN	94.95%	6.05%	0.93	0.03	0.93	0.93	0.93	0.95	9.48%

Figure 4 presents a comparison of prediction accuracy across all learning classifiers, as well as a comparison of their relative absolute errors, for your convenience. Using the image, we can tell the difference between the results generated in each of the three cases: before filtering, post SMOTE (oversampling), and post Spread subsampling (as shown in the example) (under-sampling). According to the data shown in Figure 4, the naive bayes algorithm has produced remarkable results in both instances (pre-filtering methods and post-filtering techniques). Following exposure to a spread subsampling approach, the KNN classifier was found to have significantly higher prediction accuracy than the other learning classifiers, as shown in Tables 1, 2, and 3. The KNN classifier was also found to have significantly higher prediction accuracy than the other learning classifiers. The relative absolute error has also been greatly decreased as a result of the introduction of the SMOTE and Spread subsampling methods, as previously stated. However, with the exception of random trees, where it has been shown to be aggravated when used in combination with an under-sampling method, this mistake has been reduced in almost all learning classifiers save for those based on random trees.

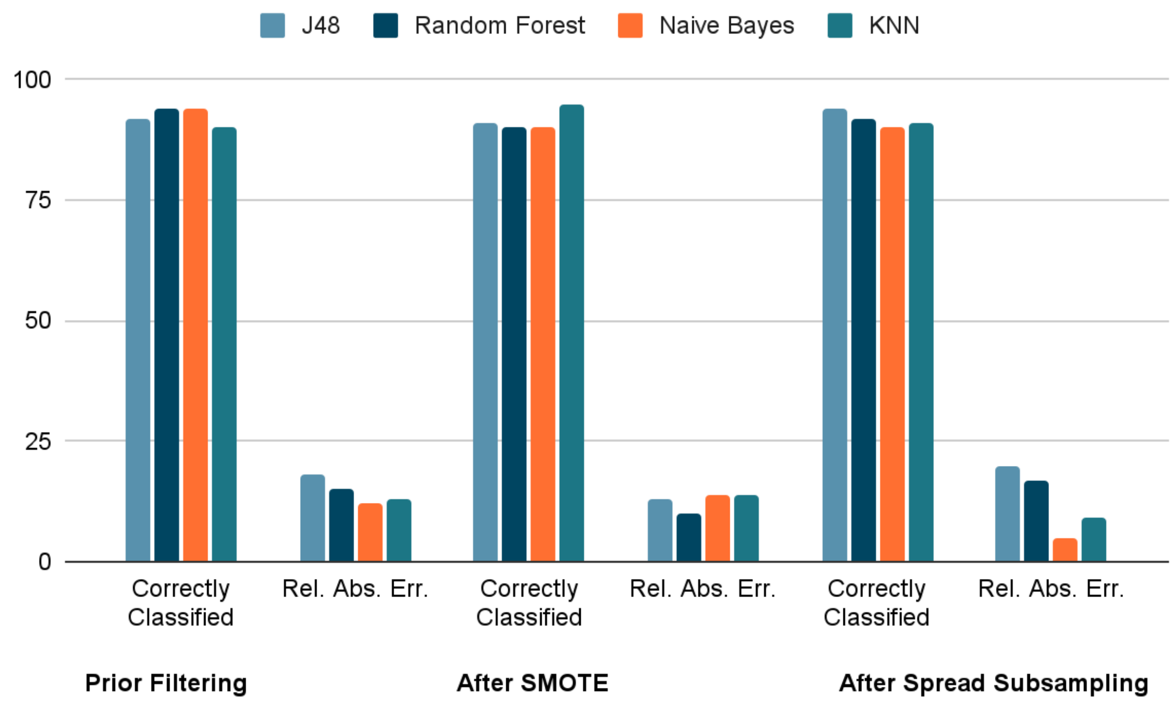

Figure. 4. Explains the variations in outcomes obtained when using a variety of strategies

3.2.1. Boosting approach

When we utilized the boosting approach to generate a comparable collection of learning classifiers that performed in the same manner on our educational dataset, we got a similar set of results. Therefore, any improvement in prediction accuracy achieved by either boosting or base classifiers using ensemble learning methodologies may be confirmed. Although we found that individual prediction models outperformed boosting paradigms, with the exception of boosting with the J48 and the Naive Bayes, we also discovered that the accuracy of classifying correct instances was boosted with 96.31%. We then investigated the model that had the least misclassification error, which turned out to be 4.77%. The outcomes of the ensemble learning process were acquired, and they are described in full in Table 5 below. Upon careful examination of the results presented in the aforementioned tables 2 and 5, it becomes clear that the relative absolute error associated with the learning base classifiers (j48 and knn) has characterized notable results when these learning base classifiers have been subjected to an ensemble learning process using boosted learning. Also revealed was that the random tree's performance reduced with the boosting strategy, from 95.77% to 90.36%, as a result of the boosting approach. This has resulted in inconsistency in the performance indicators related with the base learning classifier such as Tp rate, accuracy rate, Fp rate, recall rate, ROC area f-measure, and others when anticipating the precise instances of the base learning classifier.

Table. 5. Demonstrates the effectiveness of the boosting strategy

CN	CC	IC	TPR	FPR	Precision	Recall	F-Measure	ROC	RAE
Boost. with J48	96.33%	4.67%	0.953	0.032	0.955	0.953	0.954	0.995	7.39%

Boost. with Naive Bayes	96.05%	4.95%	0.950	0.033	0.951	0.950	0.951	0.988	10.09%
Boost. with Random tree	90.36%	10.64%	0.894	0.073	0.896	0.894	0.895	0.910	17.01%
Boost. with KNN	94.60%	6.40%	0.936	0.044	0.937	0.936	0.936	0.948	10.46%

3.2.2. Increasing the effectiveness of SMOTE

When applied to different individual based learning classifiers, such as the j48, the random tree, the knn, and the naive bayes, and when combined with the boosting strategy, the oversampling strategy, specifically SMOTE, and the boosting technique enabled us to achieve remarkable results in all of the classifiers. Base learning classifiers such as the j48 and naive bayes with boosting systems, as shown in Table 6, have shown significant improvements in prediction results when applied to the over sampling approach, as shown in this study. In J48 with boosting, the performance of the learning classifier increased from 96.31 percent to 97.41 percent; whereas, in J48 without boosting, the performance of the learning classifier increased from 96.01 percent to 97.01 percent. In this case, we use naive bayes with boosting. Both procedures yielded results that were similar to one another. However, the random tree demonstrated superior prediction accuracy over all other classifiers using the knn technique from 94.60 percent to 95.50 percent, and the random tree demonstrated superior prediction accuracy over all other classifiers using the knn technique from 94.6 percent to 95.6 percent, and the random tree demonstrated superior prediction accuracy over all other classifiers using the knn technique from 94.6 percent to 95.6 percent.

Table. 6. shows results of boosting method with SMOTE

CN	CC	IC	TPR	FPR	Precision	Recall	F-Measure	ROC	RAE
Boost. with J48	97.45%	3.55%	0.964	0.019	0.965	0.964	0.963	0.996	5.45%
Boost. with Naive Bayes	97.07%	3.93%	0.961	0.021	0.962	0.961	0.960	0.991	8.36%
Boost. with Random tree	93.04%	7.96%	0.920	0.043	0.921	0.922	0.921	0.938	12.08%

Boost. with KNN	95.50%	5.50%	0.945	0.029	0.946	0.946	0.944	0.960	8.51%

3.2.3. Increasing the impact of data after spread subsampling

In the situation of boosting employing the under-sampling technique, the outcomes of multiple base learning classifiers are shown in Table 7 in the context of boosting. According to Tables 5 and 7, increasing the number of classifiers used, such as random trees, has resulted in a decrease in performance. For example, the Tp rate has decreased from 94.60 percent to 94.00 percent. This decline in performance was found throughout the whole collection of performance estimates associated with the learning classifier, including Tp rate, Fp rate, Precision and recall, and the f-measure, among other things. Tp rate, Fp rate, Precision and recall, and the f-measure were all influenced by this decline in performance. The performance of other classifiers, on the other hand, has improved, as can be shown in the tables 5 and 7 of this report. The j48 classifier, the naive bayes classifier, and the knn classifier are examples of such classifiers.

Table. 7. This table illustrates the outcomes of the boosting approach combined with Spread Sub-Sampling

CN	CC	IC	TPR	FPR	Precision	Recall	F-Measure	ROC	RAE
Boost. with J48	96.55%	4.45%	0.955	0.22	0.956	0.956	0.954	0.996	6.87%
Boost. with Naive Bayes	92.62%	8.38%	0.916	0.042	0.917	0.916	0.915	0.937	12.67%
Boost. with Random tree	96.86%	4.14%	0.959	0.021	0.960	0.961	0.959	0.992	7.52%
Boost. with KNN	94.00%	7.00%	0.930	0.035	0.932	0.929	0.930	0.985	10.86%

Several classifiers trained using boosting techniques are illustrated in this figure, with histograms indicating their classification accuracy and relative absolute error shown in this illustration. According to the diagram below, there are three types of outcomes: boosting with each base learning classifier prior to using the filtering approach, results obtained after applying the SMOTE algorithm, and results obtained after applying the spread subsampling algorithm. Boosting with each base learning classifier prior to using the filtering approach Figure 5 shows the results of a prior application of the filtering process, in which a learning classifier such as the J48 with boosting attained the highest accuracy of 96.33 percent when compared to other learning classifiers used in the industry. In contrast, following the implementation of the SMOTE and Spread subsampling methods in each instance after the deployment of the approaches, naive bayes with boosting have obtained remarkable prediction accuracy (97.07 percent and 96.86 percent , respectively). Also evident from the figure is that the least relative error across all phases, including results obtained prior to the filtering process, results obtained following the SMOTE and Spread subsampling approaches, has been analyzed with two classifiers involving boosting concepts, namely, j48 and naive bayes, in order to determine the least relative error. j48 is a classifier involving boosting concepts that was used to determine the least relative error.

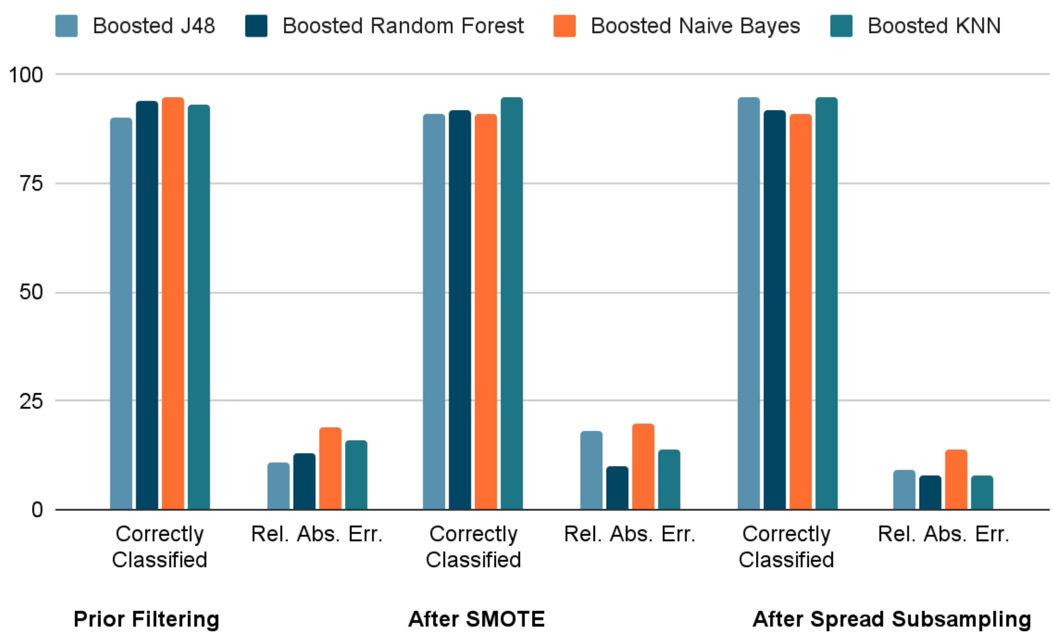

Figure. 5. It displays the outcomes of several tactics that have been used.

Figure 6 depicts the ROC curves from the output class for Division I, Division II, and Division III, respectively. Figure 6 shows the ROC curves for three classes from the output class, each with a different ROC value. The many curves constructed for each class label in Figure 6 demonstrate that the accuracy of prediction has grown significantly. Figure 6: Accuracy of prediction has increased drastically. Although the ROC curve representing Division III has shown good performance in this situation, it is because the ROC curve in this case is more dominating than in the other cases that the process has achieved its maximum potential degree of accuracy.

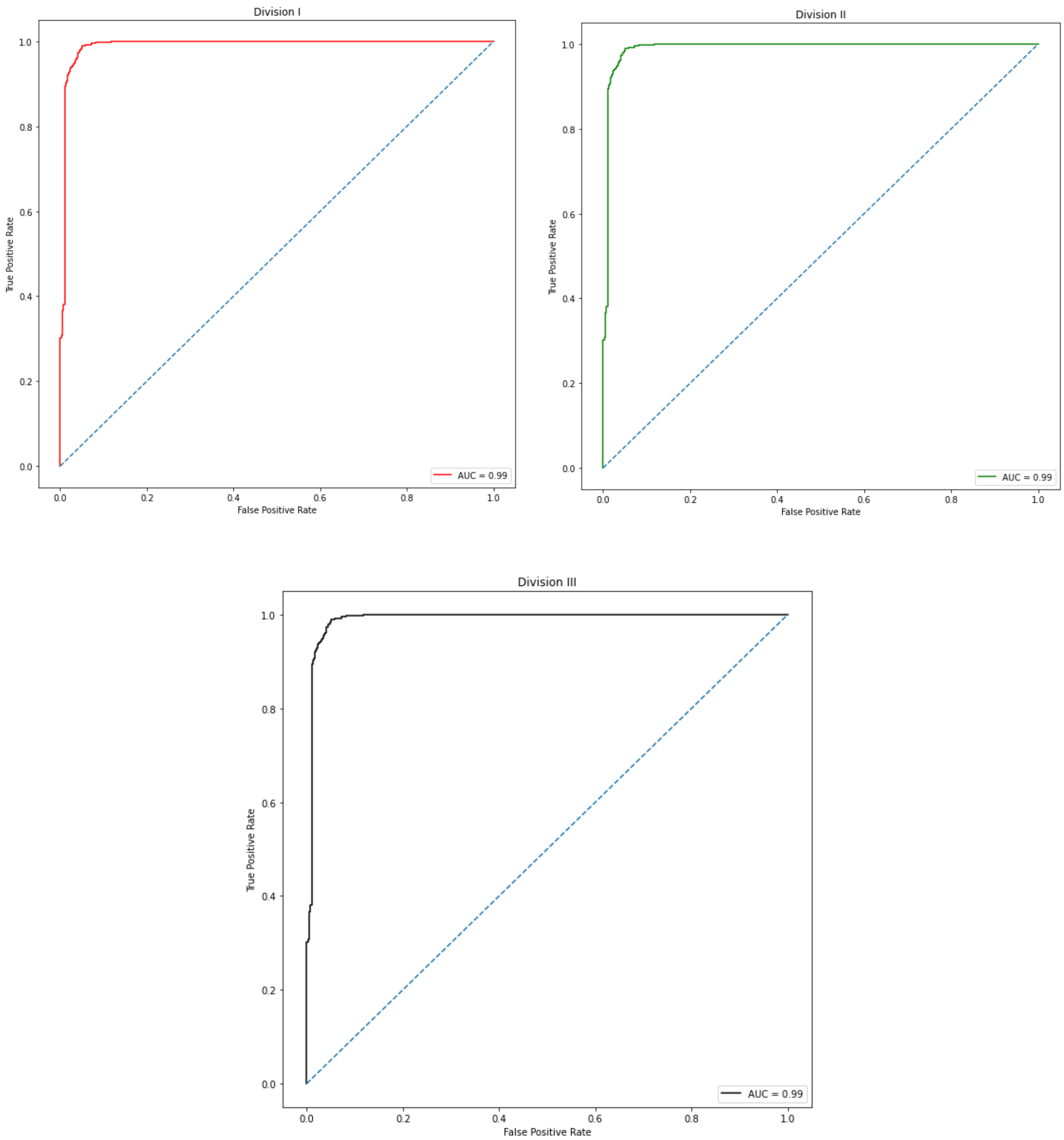

Figure. 6. The ROC curve for three different classes is shown.

Using the underlying screenshot, we can observe the predictions made by the proposed model, which are shown in table 8. Table 8 depicts a contrast between the original and predicted classes, with a few examples that have been wrongly categorized by the suggested model being drawn to the attention of the viewer.

Table. 8. The proposed system expected model output

Sample	Overall Score	Grade	
		Original	Predicted
1	77.65	II	II
2	81.22	II	II

3	44.21	~~III~~	~~II~~
4	95.98	I	I
5	70.12	II	II

4. Conclusion

In this study examination, the primary goal was to determine if filtering procedures or ensemble methods had a significant influence on the accuracy of predictions made by learning classifiers Furthermore, as a result of the results, we have developed a more accurate prediction approach for use with educational datasets. Because of this, two prediction models, called base level learning and meta level learning, have been presented in order to produce statistically significant results. When compared to all previous individual learning classifiers, the naive bayes classifier obtained a remarkable prediction accuracy of 96.51% prior to the advent of filtering or ensemble techniques, which was unprecedented at the time. In this case, the conclusions would have been erroneous and biased due to the unequal distribution of the data in the dataset. It was as a result of these findings that the researchers discovered that each learning classifier experienced a significant improvement and that among each base classifier, naive bayes achieved an impressive prediction accuracy of 98.16% when using the over-sampling technique, which was achieved using the filtering approaches. According to the findings, the knn classifier beat other learning classifiers in terms of prediction accuracy when the under-sampling technique was used to train the classifier. Researchers discovered that when they examined the ensemble approach and attempted to corroborate which method had produced better results, they discovered that boosting without being subjected to any filtering approaches, such as oversampling and undersampling procedures, demonstrated significantly higher prediction accuracy than individual classifiers. As a consequence of our findings, when we tested the ensemble approach using both under-sampling and under-sampling methodologies, prediction accuracy increased by a substantial amount in both situations as compared to when we tested it using individual classifiers.

References

[1] S. G. Suganeshwari, I. Ibrahim, and G. Li, “Lazy collaborative filtering with dynamic neighborhoods,” Inf. Discov. Deliv., vol. 46, no. 2, pp. 95–109, Jan. 2018, doi: 10.1108/IDD-02-2018-0007.

[2] W. Zhang, “Decomposition based least squares iterative estimation algorithm for output error moving average systems,” Eng. Comput., vol. 31, no. 4, pp. 709–725, Jan. 2014, doi: 10.1108/EC-07-2012-0154.

[3] F. Alyari and N. Jafari Navimipour, “Recommender systems,” Kybernetes, vol. 47, no. 5, pp. 985–1017, Jan. 2018, doi: 10.1108/K-06-2017-0196.

[4] M. Taimoor and L. Aijun, “Neural-sliding mode approach-based adaptive estimation, isolation and tolerance of aircraft sensor fault,” Aircr. Eng. Aerosp. Technol., vol. 92, no. 2, pp. 237–255, Jan. 2020, doi: 10.1108/AEAT-05-2019-0106.

[5] D. H. Zhu, Y. W. Wang, and Y. P. Chang, “The influence of online cross-recommendation on consumers’ instant cross-buying intention,” Internet Res., vol. 28, no. 3, pp. 604–622, Jan. 2018, doi: 10.1108/IntR-05-2017-0211.

[6] P. Virdi, A. D. Kalro, and D. Sharma, “Consumer acceptance of social recommender systems in India,” Online Inf. Rev., vol. 44, no. 3, pp. 723–744, Jan. 2020, doi: 10.1108/OIR-05-2018-0177.

[7] Y. Ding, X. Xiao, X. Huang, and J. Sun, “System identification and a model-based control strategy of motor driven system with high order flexible manipulator,” Ind. Robot Int. J. Robot. Res. Appl., vol. 46, no. 5, pp. 672–681, Jan. 2019, doi: 10.1108/IR-01-2019-0012.

[8] A. Cezar and H. Ögüt, “Analyzing conversion rates in online hotel booking,” Int. J. Contemp. Hosp. Manag., vol. 28, no. 2, pp. 286–304, Jan. 2016, doi: 10.1108/IJCHM-05-2014-0249.

[9] P. Virdi, A. D. Kalro, and D. Sharma, “Online decision aids: the role of decision-making styles and decision-making stages,” Int. J. Retail Distrib. Manag., vol. 48, no. 6, pp. 555–574, Jan. 2020, doi: 10.1108/IJRDM-02-2019-0068.

[10] G. Ramaswami, T. Susnjak, A. Mathrani, J. Lim, and P. Garcia, "Using educational data mining techniques to increase the prediction accuracy of student academic performance," Inf. Learn. Sci., vol. 120, no. 7/8, pp. 451–467, Jan. 2019, doi: 10.1108/ILS-03-2019-0017.

[11] S. Sedkaoui and M. Khelfaoui, "Understand, develop and enhance the learning process with big data," Inf. Discov. Deliv., vol. 47, no. 1, pp. 2–16, Jan. 2019, doi: 10.1108/IDD-09-2018-0043.

[12] J. Ranjan and K. Malik, "Effective educational process: a data‐mining approach," VINE, vol. 37, no. 4, pp. 502–515, Jan. 2007, doi: 10.1108/03055720710838551.

[13] X. Du, J. Yang, J.-L. Hung, and B. Shelton, "Educational data mining: a systematic review of research and emerging trends," Inf. Discov. Deliv., vol. 48, no. 4, pp. 225–236, Jan. 2020, doi: 10.1108/IDD-09-2019-0070.

[14] G. Özdağoğlu, G. Z. Öztaş, and M. Çağliyangil, "An application framework for mining online learning processes through event-logs," Bus. Process Manag. J., vol. 25, no. 5, pp. 860–886, Jan. 2019, doi: 10.1108/BPMJ-10-2017-0279.

[15] X. Yang, "Influence of informational factors on purchase intention in social recommender systems," Online Inf. Rev., vol. 44, no. 2, pp. 417–431, Jan. 2020, doi: 10.1108/OIR-12-2016-0360.

[16] F. Afsahhosseini and Y. Al-Mulla, "Smart, hybrid and context-aware POI mobile recommender system in tourism in Oman," J. Cult. Herit. Manag. Sustain. Dev., vol. ahead-of-print, no. ahead-of-print, Jan. 2021, doi: 10.1108/JCHMSD-08-2021-0148.

[17] Y. Yu, Z. Wang, and C. Lu, "An extended Kalman particle filter for power system dynamic state estimation," COMPEL - Int. J. Comput. Math. Electr. Electron. Eng., vol. 37, no. 6, pp. 1993–2005, Jan. 2018, doi: 10.1108/COMPEL-11-2017-0493.

[18] D. Samara, I. Magnisalis, and V. Peristeras, "Artificial intelligence and big data in tourism: a systematic literature review," J. Hosp. Tour. Technol., vol. 11, no. 2, pp. 343–367, Jan. 2020, doi: 10.1108/JHTT-12-2018-0118.

[19] A. Ray, P. K. Bala, and R. Jain, "Utilizing emotion scores for improving classifier performance for predicting customer's intended ratings from social media posts," Benchmarking An Int. J., vol. 28, no. 2, pp. 438–464, Jan. 2021, doi: 10.1108/BIJ-01-2020-0004.

[20] N. Tadi Bani and S. Fekri-Ershad, "Content-based image retrieval based on combination of texture and colour information extracted in spatial and frequency domains," Electron. Libr., vol. 37, no. 4, pp. 650–666, Jan. 2019, doi: 10.1108/EL-03-2019-0067.

[21] B. Zhang, G. Du, W. Shen, and F. Li, "Gesture-based human-robot interface for dual-robot with hybrid sensors," Ind. Robot Int. J. Robot. Res. Appl., vol. 46, no. 6, pp. 800–811, Jan. 2019, doi: 10.1108/IR-11-2018-0245.

[22] T. Guan and L. Duan, "Recovering pose and occlusion consistencies in augmented reality systems using affine properties," Sens. Rev., vol. 30, no. 2, pp. 148–158, Jan. 2010, doi: 10.1108/02602281011022751.

[23] C. G. Selvi and L. G. G. Priya, "Three-way formal concept clustering technique for matrix completion in recommender system," Int. J. Pervasive Comput. Commun., vol. 17, no. 2, pp. 167–183, Jan. 2021, doi: 10.1108/IJPCC-07-2019-0055.

[24] K. Wakil, F. Alyari, M. Ghasvari, Z. Lesani, and L. Rajabion, "A new model for assessing the role of customer behavior history, product classification, and prices on the success of the recommender systems in e-commerce," Kybernetes, vol. 49, no. 5, pp. 1325–1346, Jan. 2020, doi: 10.1108/K-03-2019-0199.

[25] S. Gul, S. Bano, and T. Shah, "Exploring data mining: facets and emerging trends," Digit. Libr. Perspect., vol. 37, no. 4, pp. 429–448, Jan. 2021, doi: 10.1108/DLP-08-2020-0078.