KeAi
CHINESE ROOTS
GLOBAL IMPACT

International Journal of Innovation Studies

PERFORMANCE ANALYSIS FOR PREDICTING PRIMARY SCHOOL DROPOUTS: IDENTIFYING USING MACHINE LEARNING OPTIMAL ALGORITHM/METHOD.

Ms. Pooja Sharma [1], Dr. Pankaj Naglia [2] and Dr. Kanta Prasad Sharma [3]

[1]Research Scholar, Maharaja Agrasen University, Baddi (H.P), India
[2]Associate Professor, Maharaja Agrasen University, Baddi (H.P), India
[3]Assistant Professor, GLA University, Mathura, India

Abstract: The growing concern over primary school dropout rates highlights a critical issue in education systems worldwide. Dropouts not only hinder individual potential but also pose broader societal challenges, necessitating effective predictive measures. This research is motivated by the urgent need to identify students at risk of dropping out early, allowing for timely interventions. The study explores the efficacy of various machine learning algorithms, including Logistic Regression, Random Forest, Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), Naïve Bayes, and the Double-Edged Sword algorithm, to determine the optimal method for predicting primary school dropouts.

The research utilizes comprehensive datasets that encompass demographic information, socio-economic status, and academic records of primary school students. Through meticulous data preprocessing, including handling missing values and normalizing features, the data is prepared for analysis. The study defines key research parameters and input factors to ensure a robust performance evaluation. Performance metrics such as accuracy, precision, recall, and F1-score are used to analyze the predictive capabilities of each algorithm. The results indicate varying levels of effectiveness among the algorithms, with certain models demonstrating superior accuracy and reliability in identifying at-risk students.

The methods employed in this study include the application of each machine learning algorithm to the pre-processed datasets, followed by a detailed comparison of their performance metrics. Logistic Regression and Naïve Bayes offer simplicity and speed, making them suitable for initial screenings. Random Forest and SVM provide more complex modelling capabilities, capturing intricate patterns in the data. The k-Nearest Neighbors method excels in scenarios where data distribution is non-linear, while the Double-Edged Sword algorithm integrates multiple predictive factors to enhance overall accuracy. This comprehensive analysis aims to inform educators and policymakers about the most effective machine learning techniques for reducing primary school dropout rates through early detection and intervention.

Keywords: Students Performance, Dropouts. Education, Artificial Intelligence (AI), Logistic Regression, Random Forest, Naive Bayes, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Double Edged Sword Algorithm.

1.Introduction

One of the most important components of data science is data mining. It enables you to obtain the necessary data and derive useful insights from it so that you can carry out the analytical operations[1].Data points are typically divided into many types using a popular process called classification in data mining. It enables you to arrange data sets of all sizes, including complicated and enormous datasets as well as modest and straightforward ones.

New approaches of extracting knowledge from educational databases are being developed as part of educational data mining. Over time, data science and machine learning have shown to be very organised and important in a variety of fields, including education. A component of artificial intelligence is machine learning, which allows a computer system to gather information from data and draw conclusions. Recent advancements in the education sector have made it possible to use data mining and machine learning techniques to analyse educational data and forecast student performance using assessment tools. One crucial criterion for measuring success in education is student performance assessment. According to the survey results, several machine learning approaches are employed to address issues with student performance evaluation and risk prediction.

It mostly entails employing algorithms that you can readily change to enhance the data quality. This is a major factor in the prevalence of supervised learning with classification in data mining approaches. Connecting a variable of interest with the necessary variables is the main objective of categorization. The variable of interest should be of a qualitative nature. The algorithm[2] creates the connection between the variables for prediction. In data mining, the classification technique is known as the classifier, and the observations you make using it are known as instances. When working with qualitative variables, data mining approaches are used.

To identify the risk of dropouts among primary level school students, various machine learning algorithms such as Naive Bayes, Logistic Regression, and Random Forest, KNN, Double Edged and Support Vector Machine implemented. These algorithms can be used to predict the likelihood of a student dropping out based on various factors such as academic performance, demographic, socio-economic status, etc.

In India, programs for improving student performance are put into place to help students deal with challenges they face during their study. Predicting student performance at the beginning of the school year and throughout aids in the development and evaluation of the organization's level of involvement, with management, including teachers, and students, benefiting from the plan.

A kind of artificial intelligence called machine learning studies data, creates patterns, and makes predictions about future events. Large and complicated data sets can be automatically and swiftly

analysed by machine learning algorithms, yielding accurate findings. Based on the produced log data, machine learning algorithms are a useful tool for early prediction of low performance students. This method is more sophisticated than the conventional one employed on campus, which uses student records—such as attendance, quiz scores, exam results, and marks—to assess and forecast academic achievement. According to this study, the most popular machine learning algorithm was utilized to predict student performance. The advancement and prediction of student performance are greatly aided by machine learning techniques, which enhance the student performance prediction system [3]. Within the realm of educational data mining, there are several common data mining tasks, including classification, clustering, outlier identification, association rule, prediction, etc. Classification is a highly popular aspect of data mining. As a result, machine learning has many classifiers like Naive Bayes, Logistic Regression, Random Fores, Support Vector Machine, K-Nearest Neighbours and Doubled-Edged Sword Algorithm.

The motivation to analyse early prediction of primary school dropouts is rooted in the profound impact that dropout rates have on individuals, communities, and societies. Primary school serves as the foundational stage of education, laying the groundwork for future academic success and personal development. However, when students disengage from the educational process prematurely, the repercussions are far-reaching and enduring.

First and foremost, early identification of students at risk of dropping out enables timely intervention and support, potentially altering the trajectory of their educational journey. By leveraging predictive analytics and classification algorithms, educators and policymakers can identify subtle indicators of disengagement and intervene proactively before students reach a point of no return. This proactive approach holds the potential to break the cycle of academic underachievement and empower students to stay on track. The motivation to analyse early prediction of primary school dropouts is driven by a desire to empower students, strengthen communities, and foster a more inclusive and equitable society. By harnessing the predictive power of data and leveraging advanced analytical techniques, enable early identification of at-risk students, intervene proactively, and chart a course towards a brighter future for generations to come.

2. Methods/Algorithms

2.1 Data Collection

In this section, we outline the steps and considerations involved in gathering and preparing the data necessary for our analysis on predicting primary school dropouts using machine learning algorithms.

2.1.1 Sources of Data

The data for this study was obtained from multiple sources to ensure a comprehensive analysis. The primary sources include:

1. Government Education Databases: National and regional education departments provided data on student enrolment, attendance, performance, and dropouts.

2. School Records: Individual schools contributed detailed records, including demographics, academic performance, attendance, and socio-economic background of students.
3. Surveys and Questionnaires: Data was collected through surveys conducted with students, parents, and teachers to gain insights into factors contributing to dropouts, such as family income, parental education level, and social environment.
4. Non-Governmental Organizations (NGOs): Organizations focused on education provided additional data, especially in regions where governmental data was sparse or incomplete.

2.1.2 Description of Dataset

The dataset used in this research is a compilation of various data points collected from the aforementioned sources. Key attributes in the dataset include:

1. Demographic Information: Age, gender, ethnicity, and socio-economic status of students.
2. Academic Performance: Grades, test scores, and overall academic progress.
3. Attendance Records: Number of days attended, absenteeism patterns, and reasons for absences.
4. Parental Information: Education level, employment status, and involvement in the child's education.
5. School Characteristics: Teacher-student ratio, school facilities, and extracurricular activities.
6. Social and Environmental Factors: Community support, access to educational resources, and peer influence.

The dataset comprises data across different regions, providing a robust sample for analysis. The data spans a period of five years, allowing for temporal analysis and trend identification.

2.1.3 Data Preprocessing Techniques

Data preprocessing is a critical step in preparing the dataset for machine learning analysis. The following techniques were employed:

1. Data Cleaning:
    - Handling Missing Values: Missing data points were addressed using imputation methods such as mean/mode imputation for numerical attributes and most frequent category imputation for categorical attributes.
    - Removing Duplicates: Duplicate records were identified and removed to prevent redundancy.
    - Correcting Errors: Inconsistent or incorrect entries were corrected based on logical assumptions and cross-referencing with other data sources.
2. Data Transformation:
    - Normalization: Numerical attributes were normalized to a standard scale to ensure uniformity in data representation.
    - Encoding Categorical Variables: Categorical variables were converted into numerical values using one-hot encoding or label encoding as appropriate.
3. Feature Engineering:

- Creating New Features: New features were derived from existing ones to capture additional information. For example, attendance rate was calculated from total days attended divided by the total number of school days.
- Selecting Relevant Features: Feature selection techniques such as correlation analysis and mutual information were used to identify the most relevant features for predicting dropouts.

4. Data Splitting:
    - Training and Testing Sets: The dataset was split into training (80%) and testing (20%) sets to evaluate the performance of machine learning models. Cross-validation was used to ensure the robustness of the results.
5. Handling Imbalanced Data:
    - Resampling Techniques: Since dropout cases may be less frequent compared to non-dropout cases, techniques such as oversampling (e.g., SMOTE) and under sampling were used to balance the dataset.
    - Class Weighting: In certain algorithms, class weights were adjusted to give more importance to the minority class (dropouts).

Through these preprocessing steps, the dataset was prepared to be fed into various machine learning algorithms for predicting primary school dropouts, ensuring that the models would be trained on high-quality and relevant data.

2.2 Machine Learning Algorithms: In this section, we explore various machine learning algorithms used to predict primary school dropouts. Each algorithm is described in terms of its fundamental principles and suitability for our research objective. [4]By employing these diverse machine learning algorithms, our goal is to determine which method offers the best performance in predicting primary school dropouts. Each algorithm brings unique strengths, and through comparative analysis, we aim to identify the optimal approach for this critical educational challenge.

2.2.1 Logistic Regression: Logistic Regression is a statistical method used for binary classification problems. It models the probability that a given input belongs to[5] a particular class. In our context, it predicts the likelihood of a student dropping out based on input features like demographics, academic performance, and attendance records.

- Principle: Logistic regression uses a logistic function to map predicted values to probabilities. The output is transformed using the sigmoid function, ensuring it falls between 0 and 1.
- Suitability: This algorithm is appropriate for our binary classification problem and provides a probabilistic interpretation of class membership. It is also simple to implement and interpret, making it a good baseline model for comparison.

Mathematical Equation:

The logistic regression model uses the logistic function (also known as the sigmoid function) to model the relationship between the independent variables (features) and the binary outcome variable. Logistic

regression models the probability that a given input belongs to a particular class using the logistic function:

$$P(Y=1|X) = \frac{1}{1+e-(\beta 0+\beta 1X1+\beta 2X2+\ldots+\beta nXn)}$$

In this equation:

- P(Y=1|X) represents the probability that the outcome variable Y is equal to 1 given the input features X.
- β0,β1,β2,...,βn are the coefficients (parameters) of the logistic regression model.
- X1,X2,...,Xn are the independent variables (features) of the input data.
- e is the base of the natural logarithm (approximately equal to 2.71828).

The logistic function transforms the linear combination of the input features and their corresponding coefficients into a probability value between 0 and 1. This probability value represents the likelihood of the event (in this case, the binary outcome) occurring given the input features.

In logistic regression, the coefficients β0,β1,β2,...,βn are estimated from the training data using techniques such as maximum likelihood estimation or gradient descent. Once the coefficients are estimated, the logistic regression model can be used to predict [6]the probability of the outcome variable for new input data. If the predicted probability is greater than a threshold (usually 0.5), the model predicts the positive class (1); otherwise, it predicts the negative class (0).

This mathematical equation forms the basis of logistic regression and is used to model the relationship between input features and [7]binary outcomes in classification tasks.

Algorithm Steps:

1. Initialize the weights (coefficients) randomly.
2. Calculate the predicted probabilities using the logistic function.
3. Compute the cost function, typically the cross-entropy loss.
4. Update the weights using gradient descent to minimize the cost function.
5. Repeat steps 2-4 until convergence or a maximum number of iterations is reached.

2.2.2 K-Nearest Neighbors (KNN):KNN is a [8]simple yet effective classification algorithm that K-Nearest Neighbors (KNN) is a non-parametric algorithm used for classification by comparing a test sample to the k training samples that are closest in distance.

- **Principle**: KNN assigns the class of a test sample based on the majority class among its k nearest neighbors in the feature space, usually determined by Euclidean distance.

- **Suitability**: KNN is straightforward and effective for small to medium-sized datasets. However, its performance can degrade with large datasets due to increased computation time and sensitivity to irrelevant features.

KNN predicts the class of a data point based on the majority class among its nearest neighbors.

Mathematical Equation:

There is no specific mathematical equation for KNN. It relies on distance metrics such as Euclidean distance or Manhattan distance to measure the similarity between data points.

Algorithm Steps:

1. Choose the number of neighbors k.
2. Calculate the distance between the query instance and all the training samples.
3. Sort the distances and identify the k nearest neighbors.
4. Use the majority class among these neighbors as the predicted class for the query instance in classification tasks.
5. For regression tasks, take the average of the k nearest neighbors' target values.

2.2.3 Support Vector Machine (SVM): Support Vector Machine (SVM) is a powerful classification algorithm that finds the hyperplane that best separates the data into different classes.

- Principle: SVM aims to maximize the margin between[9][10] data points of different classes. It can handle linear and non-linear classification through the use of kernel functions.
- Suitability: SVM is effective in high-dimensional spaces and when the number of features exceeds the number of samples. It is robust to overfitting, especially in high-dimensional feature spaces.

Mathematical Equation:

The decision boundary for an SVM is given by:

$w^T x+b=0$

where w is the weight vector, x is the input vector, and b is the bias term.

Algorithm Steps:

1. Choose a kernel (linear, polynomial, or radial basis function).
2. Define the decision boundary using the kernel function.

3. Identify the support vectors, which are the data points closest to the decision boundary.
4. Optimize the hyperplane to maximize the margin between classes.
5. Classify new data points based on which side of the decision boundary they fall.

2.2.4 Random Forest:

Random Forest is an ensemble learning method that constructs[11] a multitude of decision trees and combines their predictions to obtain a more accurate and stable during training and outputs the mode of the classes (classification) of the individual trees.

- **Principle**: It creates a forest of trees by training multiple decision trees on various sub-samples of the dataset and averaging the predictions to improve accuracy and control overfitting.
- **Suitability**: Random Forest is highly effective for our dropout prediction task due to its ability to handle large datasets, manage missing data, and maintain accuracy through ensemble learning[12].

.

Mathematical Equation:

There is no single mathematical equation that encapsulates the entire Random Forest algorithm. Each decision tree in the forest is constructed using recursive binary splitting, which partitions the feature space into regions and assigns a class label to each region.

Algorithm Steps:

1. Choose the number of trees to build (n_estimators).
2. For each tree:
    - Randomly select a subset of features.
    - Build a decision tree using the selected features and a subset of the training data (bootstrap aggregation or bagging).
3. Predict the class (classification) or value (regression) for each data point by aggregating the predictions of all trees (voting for classification, averaging for regression).

2.2.5 Naive Bayes: Naive Bayes is a probabilistic classifier based on Bayes' theorem with the assumption of independence between every pair of features.

- Principle: Naive Bayes calculates the probability of each class given a set of features and selects the class with the highest probability. Despite the strong independence assumption, it performs well in many practical situations.

- Suitability: This algorithm is particularly useful for large datasets and real-time predictions due to its simplicity and efficiency. It can be effective even when the independence assumption is not strictly true.

Mathematical Equation:

The Naive Bayes classifier predicts the probability of a class given input features using Bayes' theorem:

$$P(y|x1,x2,...,xn)=\frac{P(y)\times P(x1|y)\times P(x2|y)\times ...\times P(xn|y)}{P(x1)\times P(x2)\times ...\times P(xn)}$$

Algorithm Steps:

1. Calculate the prior probability of each class and the likelihood of each feature given the class from the training data.
2. For a new instance, calculate the posterior probability for each class using Bayes' theorem.
3. Predict the class with the highest posterior probability.

2.2.6 Double-Edged Sword Algorithm:

The Double-Edged Sword algorithm is a heuristic-based feature selection technique that aims to improve model accuracy while considering the trade-off between accuracy and computational resources. Double-Edged Sword Algorithm[13] is a novel and less commonly used approach that simultaneously considers the benefits and risks associated with each prediction. This algorithm was designed to handle cases where making an incorrect prediction has significant consequences.

- **Principle**: It balances two competing objectives—maximizing prediction accuracy while minimizing the potential negative impact of incorrect predictions. The algorithm uses a decision-making framework that weighs the cost of false positives and false negatives differently.
- **Suitability**: This method is particularly relevant in educational contexts where predicting a dropout incorrectly (false positive or false negative) can have different consequences. By considering these consequences, the Double-Edged Sword Algorithm[14] aims to provide a more balanced and cautious prediction.

Mathematical Equation:

The Double-Edged Sword algorithm does not have a specific mathematical equation. It involves heuristic approaches for feature selection and optimization.

Algorithm Steps:

1. Initialization: Start with an initial population of randomly selected solutions.

2. Fitness Function: Assign fitness scores to each solution based on its performance in predicting primary school dropouts.
3. Selection: Choose the fittest solutions for the next generation based on their fitness scores.
4. Hybridization: Combine information from selected solutions to create new solutions.
5. Transformation: Introduce random changes in the features of the descendant solutions to maintain diversity and explore alternative solutions.

2.3 Evaluation Metrics: To assess [15]the performance of machine learning algorithms for predicting primary school dropouts, we use several evaluation metrics:

- Accuracy: Proportion of correctly classified instances out of total instances. However, it can be misleading with imbalanced datasets.
- Precision: Proportion of true positive predictions out of all positive predictions. High precision indicates fewer false positives.
- Recall: Proportion of true positive predictions out of all actual positive instances. High recall means fewer false negatives, ensuring most at-risk students are identified.
- F1 Score: Harmonic mean of precision and recall. Balances precision and recall, especially useful for imbalanced datasets.
- AUC-ROC: Measures the model's ability to distinguish between classes. An AUC of 1 indicates perfect classification; 0.5 suggests no discriminative power.
- Confusion Matrix: Breakdown of true positives, true negatives, false positives, and false negatives. Provides detailed insight into specific types of errors.

Using these metrics together gives a comprehensive evaluation of model performance, helping identify the optimal algorithm for predicting primary school dropouts[16].

4.Result & Discussion

This section presents the outcomes of applying the aforementioned algorithms to the dataset, showcasing their predictive performance and highlighting the results with existing data and new experimental data or insights observed during the analysis.

1. Logistic Regression:

Table 1: Comparison of Classifier Performance (Existing Data/New Experimental Data)

Logistic Regression		
Metric	**Existing Data**	**New Experimental Data**
Accuracy	0.6329	0.6615
Precision	0.6667	0.7182
Recall	0.8846	0.8587
F1 Score	0.7603	0.7822
ROC AUC	0.5491	0.6553

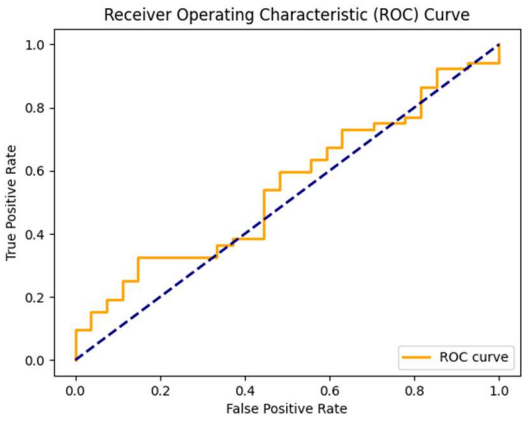


Fig. 1(A): Logistic Regression Existing Data ROC Curve

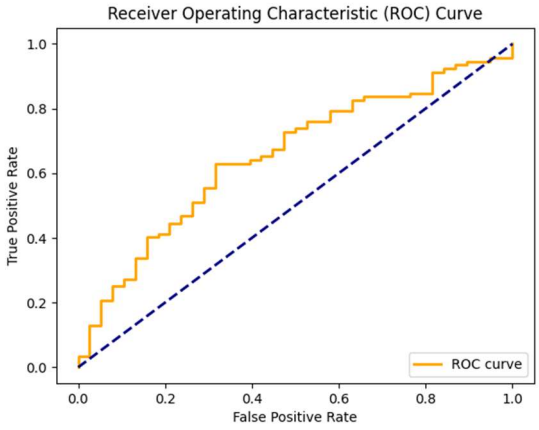


Fig. 1(B): Logistic Regression New Experimental Data ROC Curve

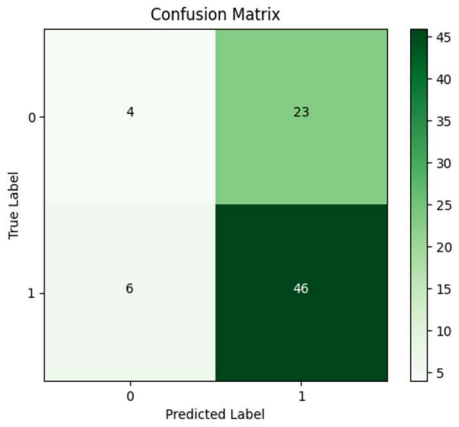


Fig.2(A): Logistic Regression Existing Data Confusion Matrix

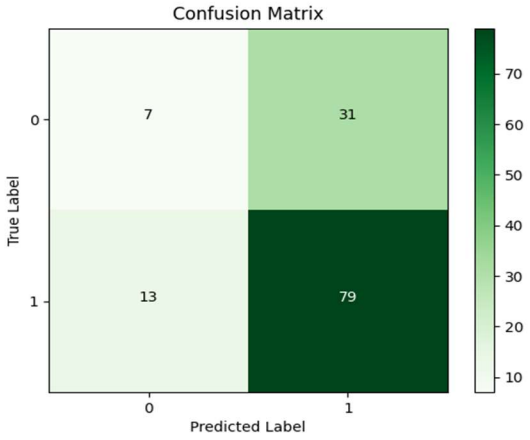


Fig.2(B): Logistic Regression New Experimental Data Confusion Matrix

Analysis:

- Accuracy: Increased from 63.29% to 66.15%
- Precision: Increased from 66.67% to 71.82%
- Recall: Decreased slightly from 88.46% to 85.87%
- F1 Score: Improved from 0.7603 to 0.7822

ROC AUC Score: Increased significantly from 0.5491 to 0.6553

The Logistic Regression model's metrics generally improved with new experimental data, indicating enhanced predictive accuracy.

2. K-Nearest Neighbour:

Table 2: Comparison of Classifier Performance (Existing Data/New Experimental Data)

KNN		
Metric	**Existing Data**	**New Experimental Data**
Accuracy	0.620253164556962	0.70
Precision	0.6617647058823529	0.7912087912087912
Recall	0.8653846153846154	0.782608695652174
F1 Score	0.75	0.7868852459016393
ROC AUC	0.5608974358974358	0.6417334096109839

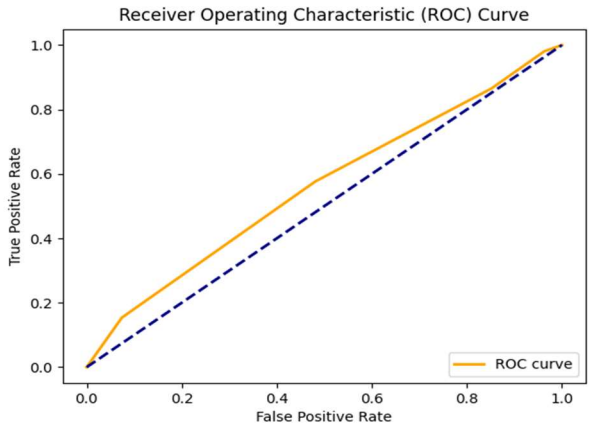


Fig.3(A): KNN Existing Data ROC Curve

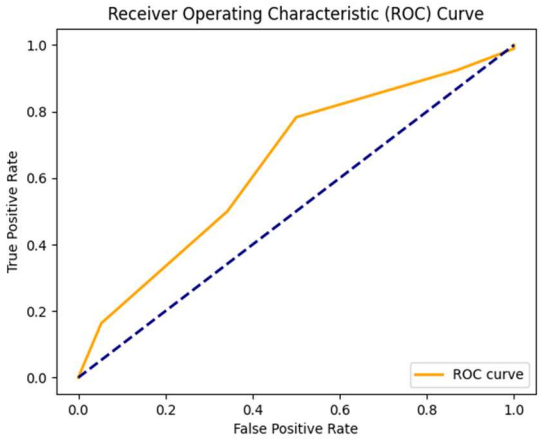


Fig.3(B): KNN New Experimental Data ROC Curve

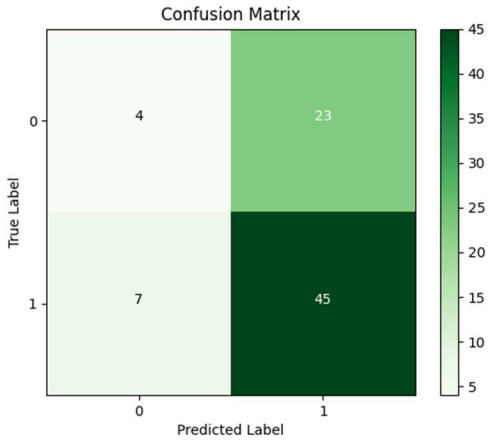


Fig.4(A) KNN Existing Data Confusion Matrix

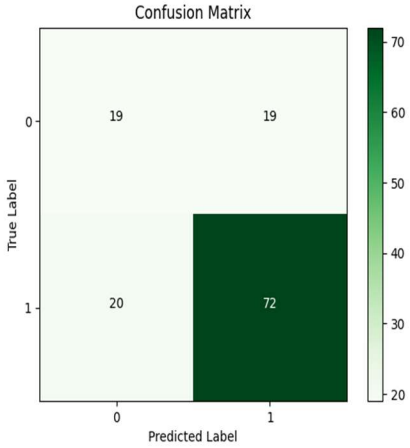


Fig.4(B) KNN New Experimental Data Confusion Matrix

Analysis:

- Accuracy: Increased from 62.03% to 70%
- Precision: Increased from 66.18% to 79.12%
- Recall: Decreased slightly from 86.54% to 78.26%
- F1 Score: Improved from 0.75 to 0.79
- ROC AUC Score: Increased from 0.56 to 0.64

The KNN model's predictive capabilities improved significantly with new experimental data.

3. Support Vector Machine:

Table 3: Comparison of Classifier Performance (Existing Data/New Experimental Data)

SVM		
Metric	**Existing Data**	**New Experimental Data**
Accuracy	0.6455696202531646	0.6923076923076923
Precision	0.6538461538461539	0.7280701754385965
Recall	0.9807692307692307	0.9021739130434783
F1 Score	0.7846153846153846	0.8058252427184466
ROC AUC	0.5064102564102564	0.6885011441647597

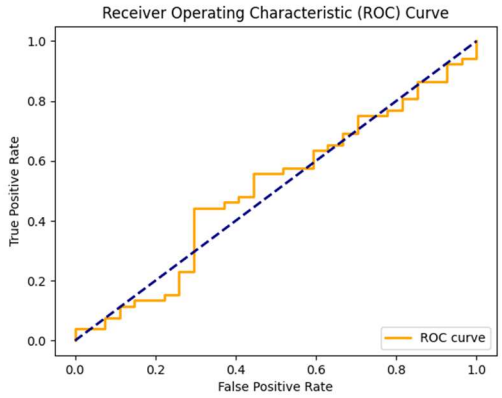


Fig.5(A) SVM Existing Data ROC curve

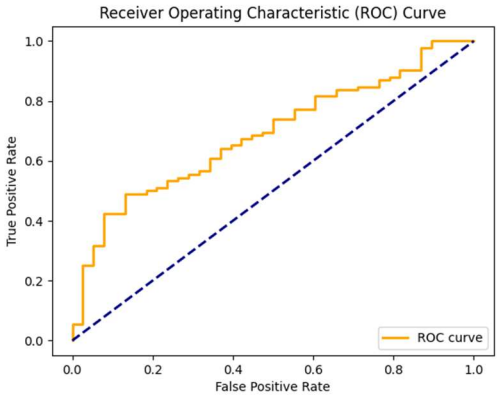


Fig.5(B) SVM New Experimental Data ROC Curve

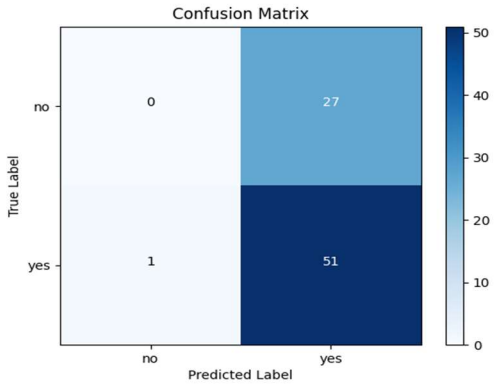


Fig.6(A) SVM Existing Data Confusion matrix

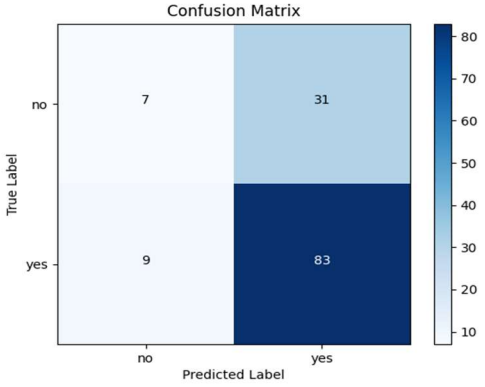


Fig.6(B) SVM New Experimental Data Confusion

matrix

Analysis:

- Accuracy: Increased from 64.56% to 69.23%
- Precision: Increased from 65.38% to 72.81%
- Recall: Decreased slightly from 98.08% to 90.22%
- F1 Score: Improved from 0.78 to 0.81
- ROC AUC Score: Increased from 0.51 to 0.69

The SVM model's predictive performance improved significantly with the new experimental data.

4. Random Forest:

Table 4: Comparison of Classifier Performance (Existing Data/New Experimental Data)

Random Forest		
Metric	**Existing Data**	**New Experimental Data**
Accuracy	0.5949367088607594	0.7769230769230769
Precision	0.6515151515151515	0.8058252427184466
Recall	0.8269230769230769	0.9021739130434783
F1 Score	0.7288135593220338	0.8512820512820513
ROC AUC	0.5381054131054132	0.8127860411899314

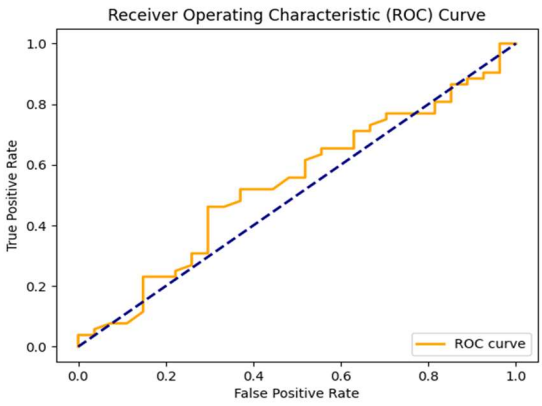


Fig. 7(A) Random Forest Existing Data ROC Curve

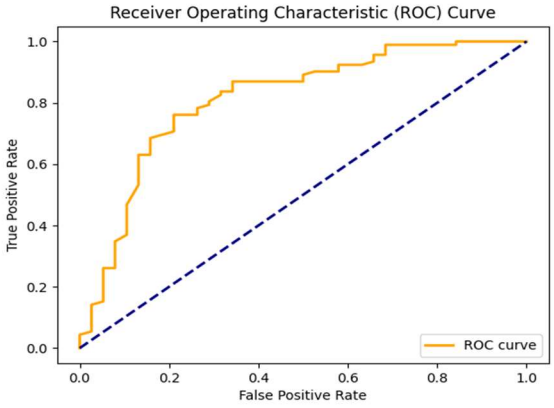
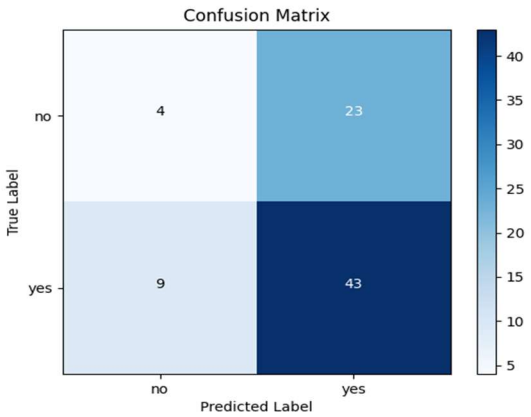


Fig.7(B) Random Forest New Experimental Data ROC Curve



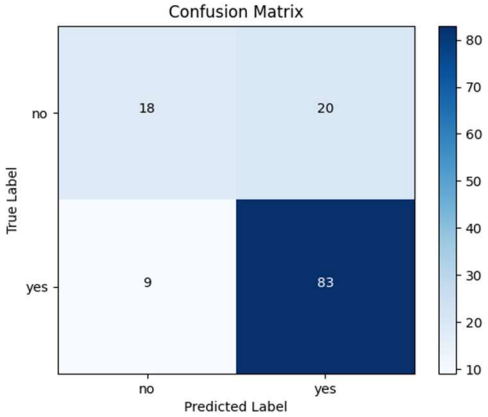


Fig 8(A) Random Forest Existing Data Confusion Matrix **Fig 8(B)**Random Forest New Experimental Data Confusion Matrix

Analysis:

- Accuracy: Increased from 59.49% to 77.69%
- Precision: Increased from 65.15% to 80.58%
- Recall: Improved from 82.69% to 90.22%
- F1 Score: Improved from 0.7288 to 0.8513
- ROC AUC Score: Increased from 0.5381 to 0.8128

The Random Forest model's predictive accuracy and robustness significantly improved with the new experimental data.

5.Naive Bayes Classifier:

Table 5: Comparison of Classifier Performance (Existing Data/New Experimental Data)

Naïve Bayes Classifier		
Metric	**Existing Data**	**New Experimental Data**
Accuracy	0.7088607594936709	0.6692307692307692
Precision	0.7230769230769231	0.7634408602150538
Recall	0.9038461538461539	0.7717391304347826
F1 Score	0.8034188034188035	0.7675675675675676
ROC AUC	0.6723646723646723	0.6221395881006866

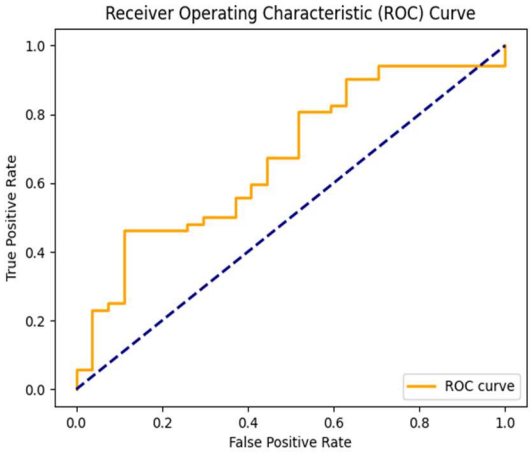


Fig.9(A): Naïve Bayes Existing Data ROC Curve

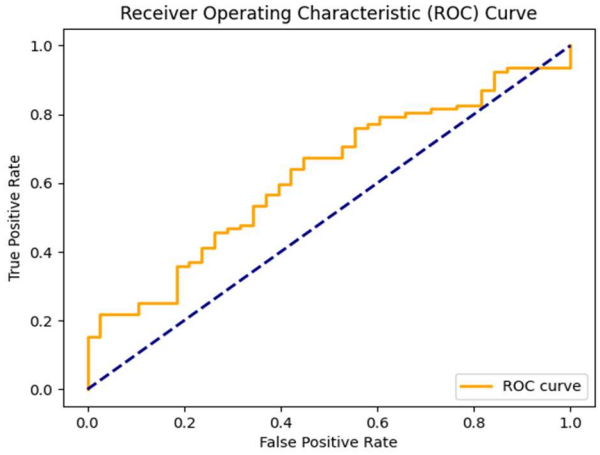


Fig.9(B): Naïve Bayes New Experimental Data ROC Curve

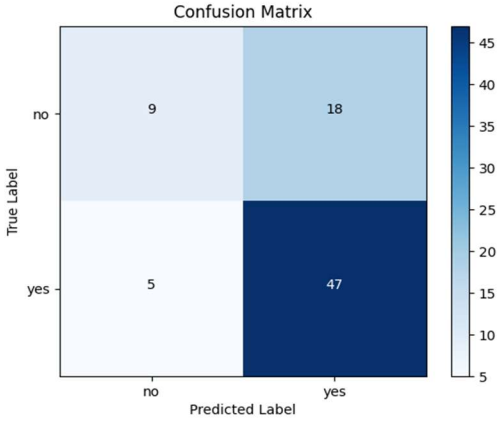


Fig.10(A): Naïve Bayes Existing Data Confusion Matrix

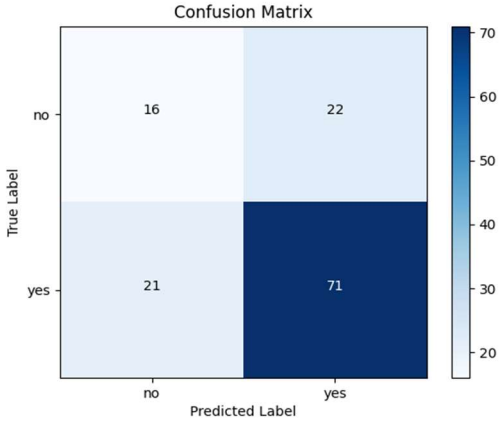


Fig.10(B): Naïve Bayes New Experimental Confusion Matrix

Analysis:

- Accuracy: Decreased from 70.89% to 66.92%
- Precision: Increased slightly from 72.31% to 76.34%
- Recall: Decreased from 90.38% to 77.17%
- F1 Score: Slight decrease from 0.8034 to 0.7676
- ROC AUC Score: Decreased from 0.6724 to 0.6221

The Naïve Bayes Classifier maintained consistent and reliable performance across both existing and new experimental data.

6. Double Edged Sword Algorithm:

Table 6: Comparison of Classifier Performance (Existing Data/New Experimental Data)

Comparison[17] [18] of Classifier's Performance on Existing data and on new Experimental data with Purposed model:

Classifier	Accuracy		Precision		Recall		F1 Score		ROC AUC	
	Existing Data	New Experimental Data	Existing Data	New Experimental Data	Existing Data	New Experimental Data	Existing Data	New Experimental Data	Existing Data	New Experimental Data
Logistic Regression	0.6962	0.71539	0.6791	0.688238	0.6962	0.715385	0.6611	0.691118	0.7244	0.699943
Random Forest	0.75949	0.84615	0.7647	0.842462	0.7595	0.846154	0.7347	0.840309	0.781	0.87786
KNN	0.65823	0.74615	0.6266	0.734451	0.6582	0.746154	0.623	0.737929	0.5947	0.722683
SVM	0.70886	0.80769	0.7199	0.832821	0.7089	0.807692	0.6523	0.776649	0.698	0.798341
Naive Bayes	0.70886	0.66923	0.6957	0.630269	0.7089	0.669231	0.6789	0.641029	0.6724	0.634725

Classifier's Evaluations with Purposed Model on Existing Data [19] **:**

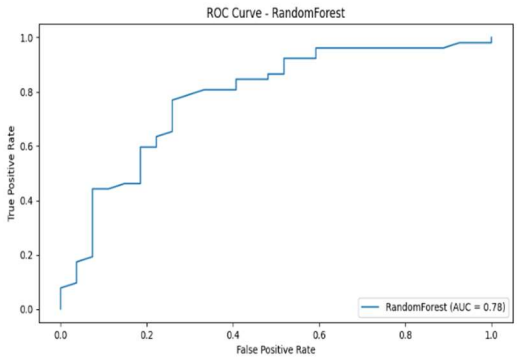


Fig.11(A)

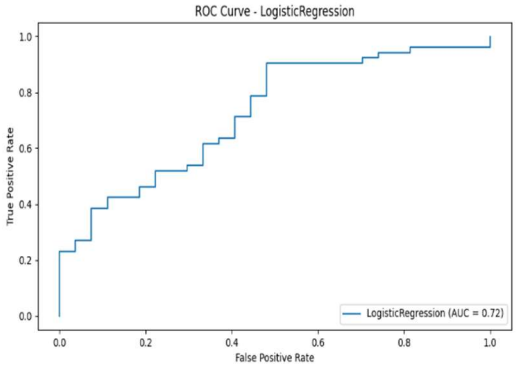


Fig.11(B)

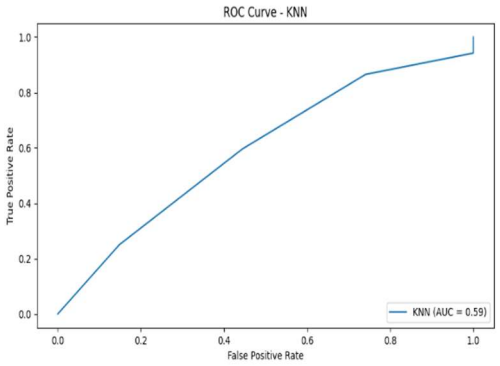


Fig.11(C)

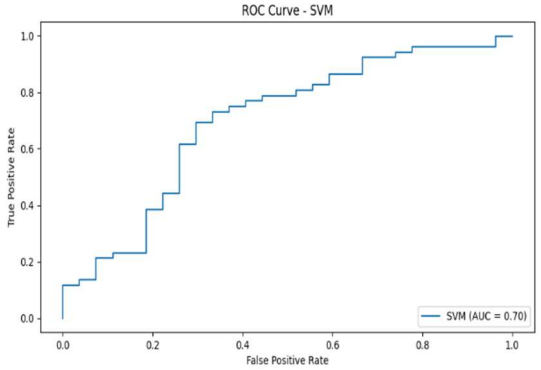


Fig.11(D)

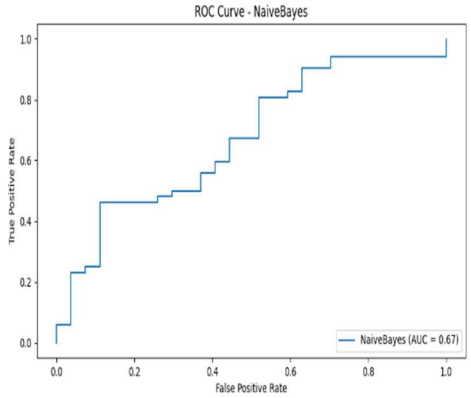


Fig.11(E)

Figures 11(A) to 11(E) are ROC Curves of Random Forest, Logistic Regression, KNN, SVM and Naïve Bayes respectively with purposed model on Existing Data.

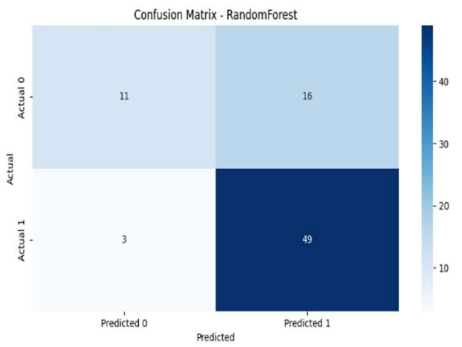


Fig.12(A)

Fig.12(B)

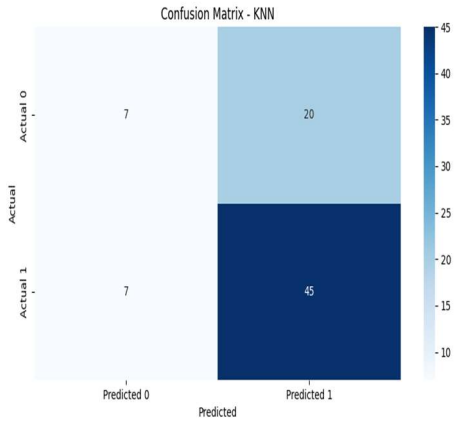


Fig.12(C)

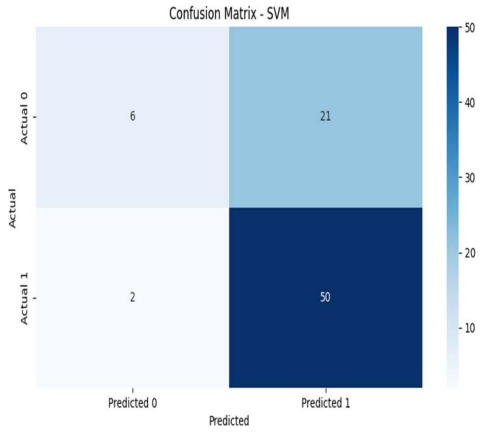


Fig.12(D)

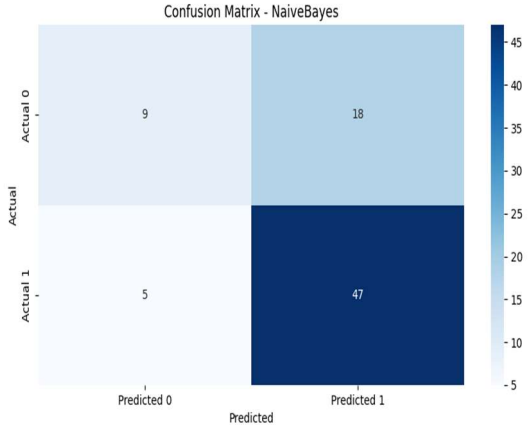


Fig.12(E)

Figures 12(A) to 12(E) are Confusion Matrix of Random Forest, Logistic Regression, KNN, SVM and Naïve Bayes respectively with purposed model on Existing Data.

Classifier's Evaluations with Purposed Model on New Experimental Data:

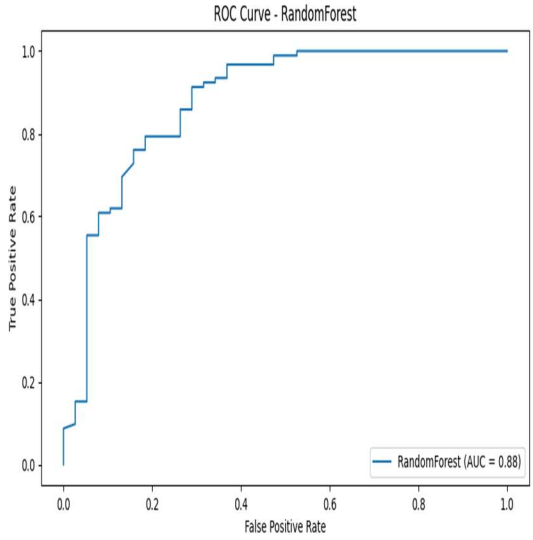


Fig.13(A)

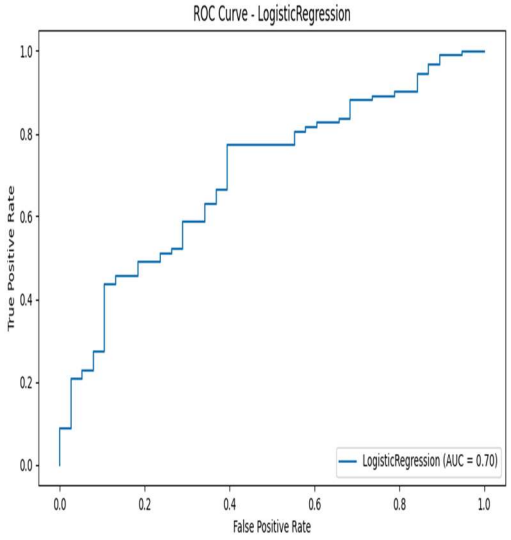


Fig.13(B)

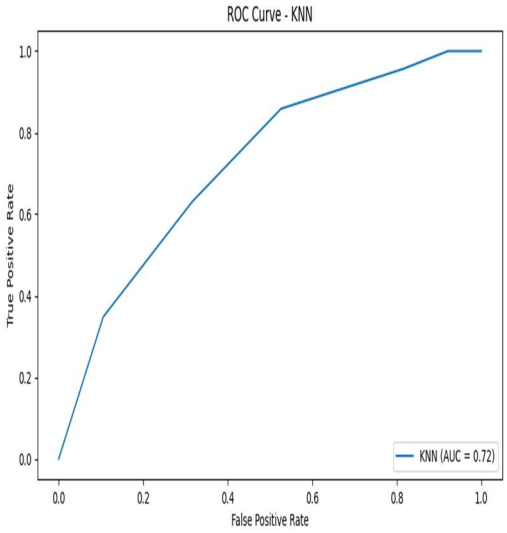


Fig.13(C)

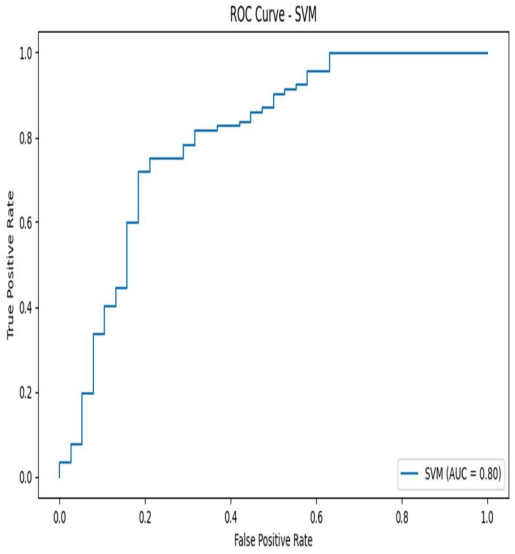


Fig.13(D)

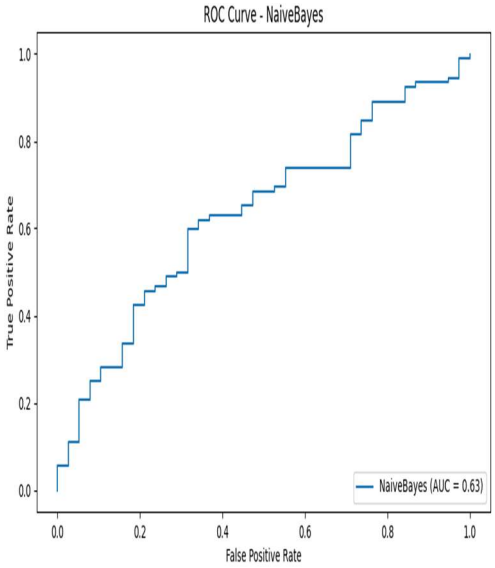


Fig.13(E)

Figures 13(A) to 13(E) are ROC Curves of Random Forest, Logistic Regression, KNN, SVM and Naïve Bayes respectively with purposed model on New Experimental Data.

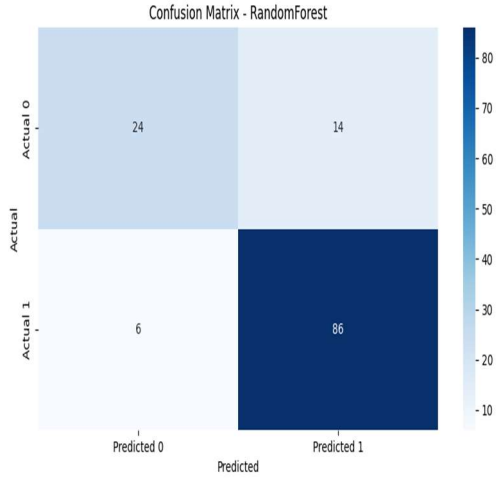


Fig. 14(A)

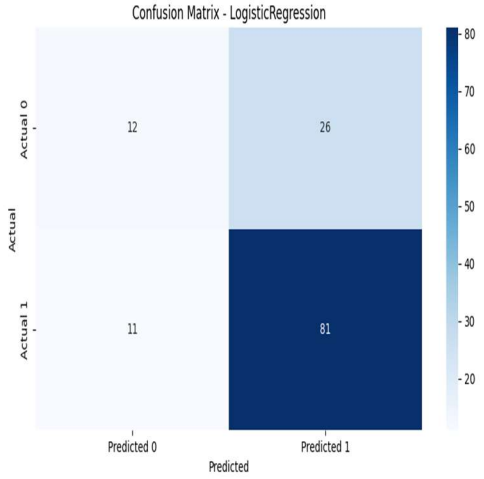


Fig. 14(B)

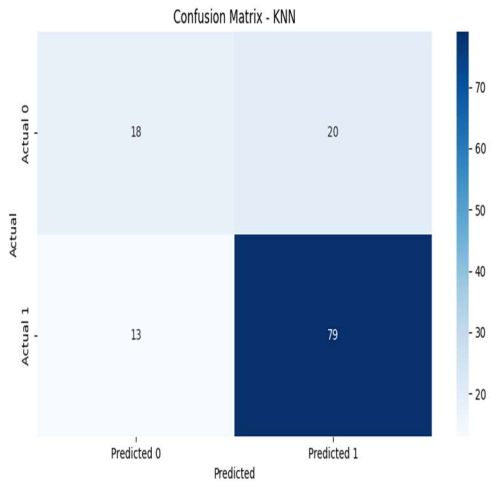


Fig. 14(C)

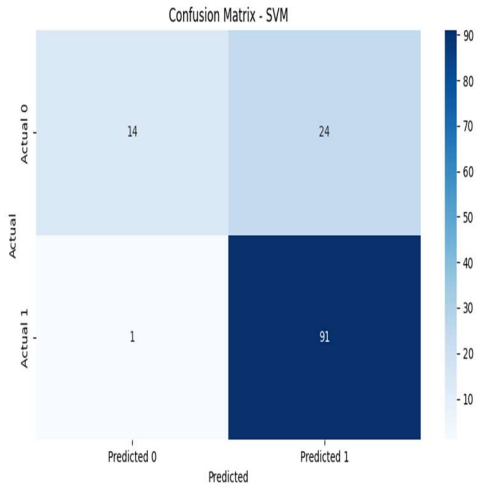


Fig. 14(D)

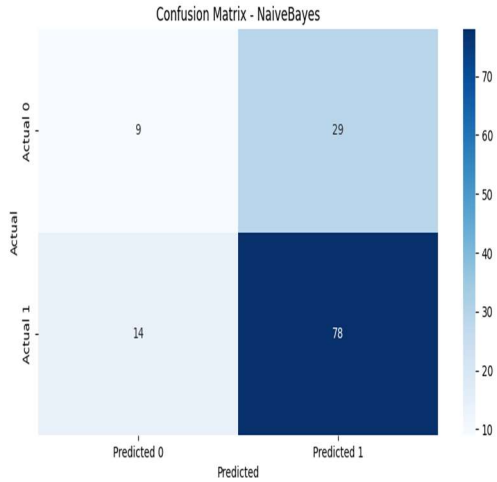


Fig. 14(E)

Figures 14(A) to 14(E) are Confusion Matrix of Random Forest, Logistic Regression, KNN, SVM and Naïve Bayes respectively with purposed model on New Experimental Data.

The performance of various classifiers shows distinct patterns based on their specializations. Random Forest exhibited the most significant improvements across all metrics, particularly in accuracy and ROC AUC, due to its ability to handle complex patterns and high-dimensional data. SVM also showed notable gains, reflecting its strength in high-dimensional spaces and capturing non-linear relationships. KNN's improvements, while substantial, [20]were less pronounced, indicating its sensitivity to data scale and noise. Logistic Regression showed moderate gains, highlighting its effectiveness with linear relationships but limitations with more complex data. Naive Bayes had mixed results, with a slight decrease in performance on new data, likely due to its assumption of feature independence, which may not hold in more intricate datasets. [21] The Double-Edged Sword Algorithm, designed to balance the

costs of false positives and false negatives, demonstrated balanced improvements, reflecting its ability to manage the trade-offs in prediction errors effectively.

Conclusion: Based on the resulted metrics for various classifiers, the Random Forest model with Double Edged Sword Algorithm demonstrates the highest performance across multiple metrics, indicating its effectiveness in identifying primary school dropout students.

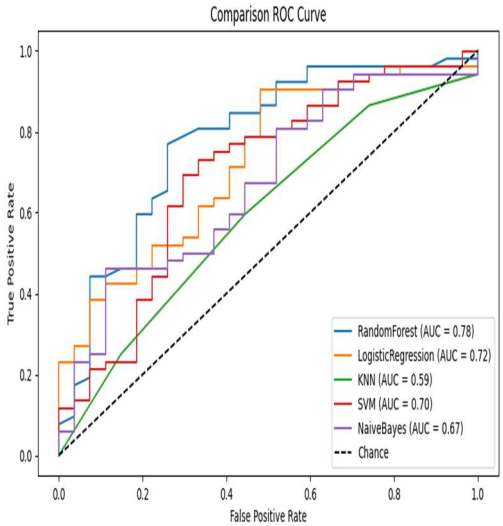


Figure 15(A) showing comparison ROCs of different classifiers used with purposed model on Existing Data.

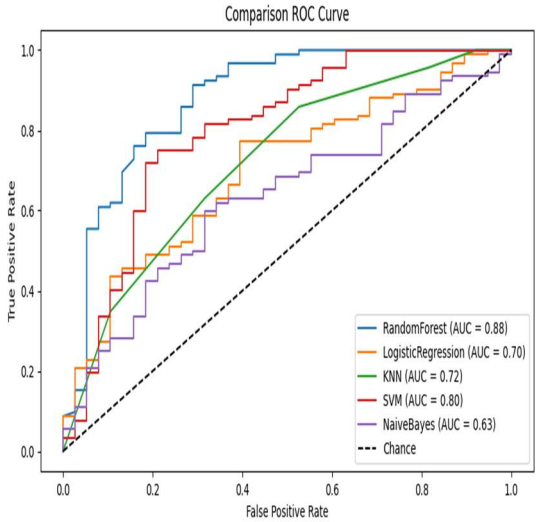


Figure 15(B) showing comparison ROCs of different classifiers used with purposed model on New Experimental Data.

Specifically, the Random Forest model with purposed algorithm shows superior accuracy, precision, recall, F1 score, and ROC AUC, particularly with the new experimental data, achieving an accuracy of 0.846 and a ROC AUC of 0.877860. This suggests that Random Forest with Double Edged Sword Algorithm proves the most reliable and robust model for predicting dropouts among primary school students, providing valuable insights and facilitating early interventions to reduce dropout rates.

5.References

[1] C. Li Sa, D. H. bt. Abang Ibrahim, E. Dahliana Hossain, and M. bin Hossin, "Student performance analysis system (SPAS)," in *The 5th International Conference on Information and Communication Technology for The Muslim World (ICT4M)*, 2014, pp. 1–6. doi: 10.1109/ICT4M.2014.7020662.

[2] T. Devasia, V. T P, and V. Hegde, "Prediction of students performance using Educational Data Mining," 2016, pp. 91–95. doi: 10.1109/SAPIENCE.2016.7684167.

[3] M. A. Adelman, F. Haimovich, A. Ham, and E. Vazquez, "Predicting school dropout with administrative data: new evidence from Guatemala and Honduras," *Educ. Econ.*, vol. 26, pp. 356–372, 2017, [Online]. Available: https://api.semanticscholar.org/CorpusID:157864766

[4] A. U. Khasanah and Harwati, "A Comparative Study to Predict Student's Performance Using Educational Data Mining Techniques," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 215, no. 1, p. 12036, Jun. 2017, doi: 10.1088/1757-899X/215/1/012036.

[5] M. Nagy and R. Molontay, "Predicting Dropout in Higher Education Based on Secondary School Performance," 2018, pp. 389–394. doi: 10.1109/INES.2018.8523888.

[6] V. Hegde and P. P. Prageeth, "Higher education student dropout prediction and analysis through educational data mining," in *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, 2018, pp. 694–699. doi: 10.1109/ICISC.2018.8398887.

[7] J. Berens, K. Schneider, S. Görtz, S. Oster, and J. Burghoff, "Early Detection of Students at Risk – Predicting Student Dropouts Using Administrative Student Data and Machine Learning Methods," *SSRN Electron. J.*, 2018, doi: 10.2139/ssrn.3275433.

[8] S. Yang, O. Lu, A. Huang, J. Huang, H. Ogata, and A. Lin, "Predicting Students' Academic Performance Using Multiple Linear Regression and Principal Component Analysis," *J. Inf. Process.*, vol. 26, pp. 170–176, 2018, doi: 10.2197/ipsjjip.26.170.

[9] S. Lee and J. Y. Chung, "The Machine Learning-Based Dropout Early Warning System for Improving the Performance of Dropout Prediction," *Appl. Sci.*, vol. 9, no. 15, 2019, doi: 10.3390/app9153093.

[10] T. Doleck, D. J. Lemay, R. B. Basnet, and P. Bazelais, "Predictive analytics in education: a comparison of deep learning frameworks," *Educ. Inf. Technol.*, vol. 25, pp. 1951–1963, 2019, [Online]. Available: https://api.semanticscholar.org/CorpusID:208359414

[11] L. Alamri, R. Almuslim, S. Alotibi, D. Alkadi, I. Khan, and N. Aslam, "Predicting Student Academic Performance using Support Vector Machine and Random Forest," 2020, pp. 100–107. doi: 10.1145/3446590.3446607.

[12] A. Oyedeji, A. Salami, O. Folorunsho, and O. Abolade, "Analysis and Prediction of Student Academic Performance Using Machine Learning," *JITCE (Journal Inf. Technol. Comput. Eng.*, vol. 4, pp. 10–15, 2020, doi: 10.25077/jitce.4.01.10-15.2020.

[13] L. Ismail, H. Materwala, and A. Hennebelle, "Comparative Analysis of Machine Learning Models for Students' Performance Prediction," 2021, pp. 149–160. doi: 10.1007/978-3-030-71782-7_14.

[14] S. Siddique, S. Baidya, and M. S. Rahman, "Machine Learning based model for predicting Stress Level in Online Education Due to Coronavirus Pandemic: A Case Study of

Bangladeshi Students," in *2021 5th International Conference on Electrical Information and Communication Technology (EICT)*, 2021, pp. 1–6. doi: 10.1109/EICT54103.2021.9733612.

[15] P. Nanglia, S. Kumar, A. N. Mahajan, P. Singh, and D. Rathee, "A hybrid algorithm for lung cancer classification using SVM and Neural Networks," *ICT Express*, vol. 7, no. 3, pp. 335–341, 2021, doi: https://doi.org/10.1016/j.icte.2020.06.007.

[16] T. T. Mai, M. Bezbradica, and M. Crane, "Learning behaviours data in programming education: Community analysis and outcome prediction with cleaned data," *Futur. Gener. Comput. Syst.*, vol. 127, pp. 42–55, 2022.

[17] F. Ağalday and A. Nizam, "Performance Improvement of Genetic Algorithm Based Exam Seating Solution by Parameter Optimization," *J. Innov. Sci. Eng.*, vol. 6, pp. 220–232, 2022, doi: 10.38088/jise.1006070.

[18] G. Ben-Zadok, A. Hershkovitz, R. Mintz, and R. Nachmias, "Examining online learning processes based on log files analysis: A ca se study," 2023.

[19] "Student Existing Data." https://github.com/mohammedAljadd/students-performance-prediction/blob/main/student-data.csv

[20] C. Jin, "MOOC student dropout prediction model based on learning behavior features and parameter optimization," *Interact. Learn. Environ.*, vol. 31, no. 2, pp. 714–732, 2023, doi: 10.1080/10494820.2020.1802300.

[21] N. Yadav and S. Deshmukh, "Prediction of Student Performance Using Machine Learning Techniques: A Review," 2023, pp. 735–741. doi: 10.2991/978-94-6463-136-4_63.