INTERNATIONAL JOURNAL OF
TECHNICAL AND NATURAL SCIENCES

DOI: https://doi.org//10.69648/XAPH5884

International Journal of Natural and Technical Sciences (IJTNS), 2025; 5(2): 17-44.

ijtns.ibupress.com

Online ISSN: 2671-3519

Application: 02.10.2025

Revision: 18.10.2025

Acceptance: 25.12.2025

Publication: 31.12.2025

Saxena, M., & Singh, N. P. (2025). Analysis of an educational dataset using a classification algorithm in conjunction with wrapper feature selection methods. International Journal of Natural and Technical Sciences, 5(2), 17-44. https://doi.org//10.69648/XAPH5884

@

Mamta Saxena

School of Engineering and Technology, MVN University Palwal, Haryana, India, https://orcid.org/0009-0007-8476-2836.

Netra Pal Singh

School of Engineering and Technology, MVN University Palwal, Haryana, India, https://orcid.org/0000-0002-3006-9522.

BALKAN JOURNALS

Analysis of an Educational dataset using Classification Algorithm in Conjunction with Wrapper Feature Selection Methods

Mamta Saxena, Netra Pal Singh

Abstract

Feature selection plays a critical role in improving the efficiency, accuracy, and interpretability of machine learning models, particularly when dealing with high-dimensional datasets. Among various approaches, wrapper-based feature selection methods are known for their ability to capture feature interactions by directly optimizing model performance. This study presents a comprehensive comparative analysis of six wrapper feature selection techniques—Recursive Feature Elimination (RFE), Sequential Forward Selection (SFS), Sequential Backward Selection (SBS), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Differential Evolution (DE)—in conjunction with five widely used classification algorithms: Decision Tree, K-Nearest Neighbour, Random Forest, Logistic Regression, and Support Vector Machine. Experiments are conducted on an educational dataset comprising 395 student records with 30 attributes obtained from the UCI repository, using different feature subset sizes (all features, top 20, top 15, and top 10). Model performance is evaluated using accuracy, precision, recall, F1-score, and AUC. The results demonstrate that wrapper methods significantly enhance classification performance while reducing dimensionality, with GA and RFE consistently emerging as the most effective techniques across multiple classifiers. DE also shows strong performance, particularly with Logistic Regression and Random Forest, whereas PSO generally underperforms in terms of AUC. Furthermore, reducing the feature set does not adversely affect predictive accuracy and, in several cases, leads to improved generalization. The findings confirm the effectiveness of wrapper methods for educational data mining and provide practical insights for selecting optimal feature–classifier combinations.

Keywords: Classification Algorithms, Feature Selection, Performance Metrics, Wrapper Methods

Introduction

Excellent academic records are constantly needed by prestigious universities, and their students are their most valuable asset. The primary focus of universities is student performance, which serves as the foundation for producing top-notch graduates and post-graduate students who will lead their countries and assume responsibility for the social and economic advancement of society. Furthermore, because it directly affects the hiring process and subsequently staff productivity, market companies are primarily concerned with university and student performance. Binmat et. al. (2014) discussed that graduates who work hard during their academic careers are able to meet the demands of their employers. The curriculum and learning assessments are used to gauge student success. Feature selection helps address this challenge by identifying the most relevant attributes for prediction. Among existing approaches, wrapper-based feature selection methods are particularly effective because they evaluate feature subsets directly using classification models. Although computationally intensive, these methods often yield more accurate and reliable predictions compared to traditional statistical techniques.

In this work, six wrapper feature selection methods are systematically evaluated in combination with five commonly used classification algorithms to predict student academic performance. The analysis is performed using different feature subset sizes to assess the trade-off between dimensionality reduction and predictive accuracy. The study aims to identify robust feature–classifier combinations that can support effective decision-making in educational data mining applications.

This paper provides a comprehensive analysis of wrapper methods. The key objectives of the research paper are the comparative evaluation of wrapper feature selection methods and the analysis of computational efficiency and model performance. The following is the paper's outline: The literature on the feature selection method related to the wrapper method is presented in Section II. The paper's research approach is presented in Section III. The discussions and results are shown in Sections IV and V. Section VI contains a description of the study's conclusion.

Literature Review

This section presents a review of the literature on Wrapper methods of Feature Selection and Classification Algorithms used to classify the instances in conjunction with wrapper methods.

Review of Literature Using Wrapper Methods

Techniques for feature selection can be broadly divided into three categories: filter, wrapper, and embedding. While integrated approaches integrate selection into the learning algorithm, filters rank features according to statistical measures. Wrappers use model performance metrics to assess subsets of features.

Several surveys, including Chandrashekar and Sahin (2014), have reviewed the taxonomy and applications of feature selection methods, highlighting their critical role in reducing overfitting and improving model interpretability. Dash and Liu (1997) emphasized the importance of integrating domain knowledge to enhance feature selection processes. As Li et al. (2017) discussed, recent advances have focused on scalable methods to handle large datasets.

Moslehi & Haeri (2020) proposed a hybrid filter-wrapper technique for feature subset selection that combines particle swarm optimization (PSO) with evolutionary-based genetic algorithms (GA). The suggested method's primary goal is to minimize the amount of time spent searching and the complexity of calculations in order to find the best answer to the feature selection problem for high-dimensional datasets.

Similarly, to balance efficiency and performance, Tang et al. (2014) highlighted the importance of hybrid approaches that integrate filter and wrapper methods. Maldonado and Weber (2009) proposed that wrapper algorithms can computationally enhance the performance of feature selection when combined with Filter Selection methods.

Singh and Karthikeyan (2024) determined that the performance of Ant colony optimization with random forest (ACO-RF) is superior for feature selection. In order to predict university student dropout, the ACO-RF wrapper technique for feature selection is suggested in research. Neural networks and machine learning methods are eventually used to validate the feature that ACO-RF chose. With a 94% accuracy rate, the neural network outperformed competing machine learning techniques.

Two-stage feature selection procedures using machine learning techniques were presented by Patel et al. (2024). The sequential backward feature selection approach is used in the second stage to show the dataset's correctness, while the wrapper method is used in the first stage to choose a combination of feature subsets from the dataset.

The classifier selection has a significant impact on the quality of the feature subset that is produced when creating a wrapper feature selection model. In a recent study, Xue et. al. (2015) conducted significant experiments to focus on the computational elements of wrappers using several classifiers. Xue et. al. (2016) carried out an interesting survey of feature selection strategies emphasizing evolutionary techniques and their salient features, such as the fitness function design, benefit mechanisms, and model representation.

Review of Literature on Classification Algorithms Used in Conjunction With Wrapper Methods

By repeatedly training a model on several subsets and assessing performance using a user-defined criterion (such as accuracy or F1-score), wrapper approaches find the optimum feature subset. Typical wrapper methods consist of: (i) Recursive Features Elimination (RFE), (ii) Step Forward Selection (SFS), (iii) Step Backward Selection (SBS), (iv) Genetic Algorithm (GA), (v) Particle Swarms Optimization (PSO), (vi) Differential Evolution (DE). These are frequently used in conjunction with classifiers such as Random Forests, Decision Trees, K-Nearest Neighbors, Support Vector Machines, and Logistic Regression.

Applying RFE with Random Forest to student datasets, Feng & Xu (2019) discovered that it enhanced model interpretability, which in turn improved dropout prediction. When comparing SFS with SBS in pattern recognition, Jain and Zongker (1997) discovered that forward selection typically outperforms SBS for sparse datasets.

Asif et al. (2017) achieved notable accuracy gains in their selection of factors influencing student academic performance in educational datasets by combining SFS with Decision Trees. In their evaluation of GA-based feature selection, Xue et al. (2016) demonstrated how well it traverses the search space, particularly when used with classifiers such as SVM and RF. Khan and Jawaid (2020) enhanced AUC scores by using GA with Random Forest to choose the best attributes for identifying students who were likely to fail.

The ability of PSO to handle huge feature spaces with less computation than GA was highlighted by Chandrashekar and Sahin (2014). Tomasevic et al. (2019) enhanced early student dropout prediction by using PSO on academic data. Using DE with classification models on biological and educational datasets, Islam et al. (2018) demonstrated faster convergence and competitive accuracy. DE has been less popular in educational contexts, but its use is expanding because of its ease of use and capacity to search globally.

Research Methodology

In this Research paper, various wrapper methods are applied on the dataset namely, Forward Selection, Backward Selection, Recursive Feature Elimination (RFE), Particle Swarm Optimization (PSO), Differential Evolution (DE) and Genetic Algorithm (GA) by taking all features and then by selecting Top 20, Top 15 and Top 10 features to evaluate the dataset by using different wrapper methods and then calculating the performance metrics for the same with different Classification Algorithms namely Logistic Regression, Decision Tree, Random Forest, SVM and KNN. The dataset is taken from the UCI data repository, which consists of 30 attributes/ features and a target variable christened as "Passed". The data is partitioned into two parts, i.e., 80 % training data set and 20% testing data set.

Objective/Questions

The main research objective is to examine the impact of wrapper feature selection algorithms on the effectiveness of classification algorithms applied to the best set of predictors. The following research questions will be addressed by this study:

Research Question 1: How do different wrapper-based feature selection methods (RFE, SFS, SBS, GA, PSO, and DE) affect the predictive performance of classification models when applied to educational datasets?

Research Question 2: Which wrapper feature selection techniques consistently identify the most informative features for predicting student academic performance?

Research Question 3: How does the choice of classification algorithm (Decision Tree, KNN, Random Forest, Logistic Regression, and SVM) influence the effectiveness of wrapper-based feature selection methods?

Research Question 4: What is the impact of reducing feature dimensionality (top 20, top 15, and top 10 features) on model accuracy, robustness, and generalization performance?

Dataset Description

The dataset comprises 395 student records with 30 features for each record. This dataset has been used in many studies and is available publicly on many data repositories such as Kaggle. It was previously used to check the students' academic success and passing rates. There are three categories of attributes in this dataset (i) demographic features (sex, age, address, family size, Parent status, health), (ii) academic background features (school, study time, failures, school support, paid, activities, nursery, higher studies, absences) and (iii) social-economic features (Mother's education, Father's education, Father's job, Mother's job, family support, reason, guardian, travelling time, internet used, romantic, family relation, free time, gout for outing, Weekday alcohol consumption, Weekend alcohol consumption). In the previous research paper, the dataset was used to study the impact of feature selection by using seven classification algorithms, and it was concluded that by selecting different features, the behavior of classification algorithms remains almost the same with all features, as well as with the Top 10 and Top 8 features.

Wrapper Methods

Wrapper methods utilize predictive models to assess feature subsets. These methods are computationally intensive but often yield superior results due to their consideration of feature interactions. Notable techniques include:

Recursive Feature Elimination (RFE): Recursively eliminating the least significant features and using the remaining features to construct models is how RFE works. It removes features gradually and assigns a feature priority based on how much it contributes to the model's performance. Through this procedure, overfitting decreases, model accuracy improves, and interpretability is enhanced. Guyon and Elisseeff (2003) introduced RFE, demonstrating its effectiveness in identifying informative genes in bioinformatics. The algorithm and formula of RFE are given in the following.

Algorithm (Guyon et al., 2009):

Step1: Train the model on the full feature set.

Step2: Compute the importance of features.

Step3: Remove the least important feature.

Step4: Repeat until the desired features remain.

Formula (Kohavi & John,1997)

Let $F=\{f_1,f_2,\ldots,f_n\}$ be the feature set.
For every step:
Train model M on F
Rank features using Importance, let say ()
Eliminate feature that are least important by using f_k=argminImportance(fi)

Genetic Algorithms (GA): By simulating natural selection, the Genetic Algorithm (GA) effectively finds the ideal subset of traits. In order to evolve feature subsets that optimize model performance while reducing redundancy, GA uses selection, crossover, and mutation. Holland (1992) described the foundational principles of genetic algorithms, which have since been applied to feature selection for optimizing search spaces, as detailed by Siedlecki and Sklansky (1989). The algorithm of GA is as follows:

Algorithm (Holland, 1992):
Step1: Generate initial population of chromosomes $X \in \{0,1\}^n$
Step2: Evaluate fitness: $f(X_i)$
Step3: Select parents via tournament or roulette
Step4: Crossover:
X_{new} =crossover (X_i, X_j)
Mutation:
$X_{new}[k] = 1 - X_{new}[k]$ with probability p_m
Step5: Form a new population and repeat (Khan & Jawaid, 2020; Siedlecki & Sklansky,1989)

Step Forward Selection (SFS): SFS follows a greedy approach by gradually identifying the most pertinent features and adding them one at a time according to their contribution to model performance. Sequential Forward Selection (SFS) is utilized in wrapper methods for feature selection. It lessens computational complexity while enhancing accuracy. Kohavi and John illustrated that using an iterative process, we begin with a blank set of features and continue to add features that enhance our model the most with each iteration. The stopping criterion is when the performance of the model does not increase with the addition of a new variable, as discussed by Siedlecki and Sklansky (1989). The algorithm and Formula of SFS are as follows:

Algorithm [Guyon and Elisseeff, 2003]:

Step1: Initialization: Let the original data set is $F= \{f_1,f_2,\ldots, f_n\}$. Initialize with an empty feature set, let's say $S=\emptyset$

Step2: **Candidate Feature Selection**: For each feature $f \notin F\setminus S$, train a model using a feature subset $S \cup \{f\}$ and evaluate model performance with $J(.)$ (e.g., accuracy, Precision, AUC, etc.)

Step3: Feature Selection: Select the feature f^* that maximizes the elevation matrix as

$$f^* = \underset{f \in F \backslash S}{arg\max} \left(J\left(S \cup \{f\}\right)\right)$$

Step 4: Update the feature subset:

S = S {f*}

Step 5: if |S| =k (desired number of features or performance improved is less than a limit stop

Otherwise

Go to Step 2

Formula:

$$f* = \underset{f \in F \backslash S}{\text{argmax}}\ J\ (S \cup \{f\})$$

Where J is performance functions, i.e., accuracy, F1-Score, etc. (Pudil et. al., 1994). F\S .is F minus S.

Step Backward Selection (SBS): Sequential Backward Selection (SBS) follows the Dimensionality Reduction technique for eliminating the least significant features to enhance model performance. SBS is utilized in wrapper approaches for feature selection. To make sure that only the most essential features are left, it begins with the entire list of features and removes each one individually. The criterion for terminating is until the removal of the feature results in no discernible improvement in the model's performance, as discussed by Kohavi and John (1997). The algorithm and Formula of SBS are as follows:

Algorithm (Jain & Zongenkar, 1997):

Step1: **Initiation:** Start with the full feature set S=F

Step2: **Evaluation:** For each feature f ∈ S, evaluate J (S\ {f})

Step3: **Elimination:** Remove features that least affect performance.

Step4: **Iteration:** Elimination: Repeat steps 2-3 until k features remain.

Formula:

$$f* = \underset{f \in S}{\text{argmax}}J$$ (Guyon and Elisseeff,2003)

Particle Swarm Optimization (PSO): The optimization method known as Particle Swarm Optimization (PSO) is modeled after the social behavior of fish schools or flocks of birds. Since its introduction by Kennedy and Eberhart (1995), it has been extensively used to solve a variety of optimization issues, such as combinatorial and continuous optimization. PSO is utilized for feature selection in wrapper methods as it effectively identifies the most appropriate subset of features by modeling a swarm's behavior. In order to improve model performance and decrease redundancy, it strikes a balance between exploration and exploitation. The algorithm and Formula of PSO are as follows:

Algorithm (Waheed et al., 2020):
Step1: **Initialize** particles with random positions $\in \{0,1\}^n$
Step2: **Evaluate** fitness f () (e.g., model accuracy)
Step3: **Update velocity**: $v_i = w\, v_i + (pbest_i - x_i) + (gbest - x_i)$
Step4: **Update position**: $x_i = \text{sigmoid}(v_i) > rand$
Step5: **Update** pbest and gbest

Repeat until max iterations (Kennedy & Eberhart,1995)

Differential Evolution (DE): Another population-based optimization technique that was first presented by Storn and Price (1997) is called Differential Evolution (DE). In contrast to PSO, DE is predicated on the idea that improved solutions can be evolved through the recombination and mutation of individuals within the population. It works very well for optimization issues with real values. To determine the best choice, DE looks at a large number of feature subsets. It works effectively with rich in features datasets. Finding the ideal feature combination to enhance model performance and managing high-dimensional datasets are two areas where DE excels. DE improves predictive performance through the selection of the most pertinent features. The algorithm and Formula of DE are as follows:

Algorithm (Storn & Price, 1995):
Step1: Initialize population $X_i \in [0,1]^n$
Step2: Mutation:$V_i = X_{r1} + F \cdot (X_{r2} - X_{r3})$
Ui[j]={Vi[j] if r and j<CR
Xi[j]otherwise
Step3: Selection:
Xi= {U_i if $f(U_i) > f(X_i)$
Xi otherwise
Step 4: Repeat until convergence [Das and Suganthan,2011] [Sharma and Kaur, 2020]

Kohavi and John (1997) highlighted the trade-off between computational cost and selection accuracy in their novel work on wrapper techniques.

Wrapper Methods in Conjunction with Classification Algorithm

RFE Pseudocode for Classification Algorithm (Guyon et. al. 2002; Pedregosa et. al. 2011)

Input: Take D as the dataset, C as the classifier, and k as the number of features to select.

Output: Select Specific features $F_{selected}$
F –For all features
Continue until |F| == k.
Use features to train classifier C on the dataset. F
Calculate feature importance scores (such as Gini importance or coefficients) from C.
Take away F's least significant feature.
$F_{min} = \arg\min_{f \in F} I(f)$
Return $F_{selected}$ = F at the end

SFS Pseudocode for Classification Algorithm (Jain & Zongekar, 1997; Guyon & Elisseeff, 2003; Pudil et al., 1993)

Input: Take D as the dataset, C as the classifier, and k as the number of features to select.

Output: Let selected features be $F_{selected}$
$F_{selected} \leftarrow \{\}$
While $|F_{selected}| < k$ do
best_score ← -∞
For each feature, f not in F_selected:
$F_{temp} \leftarrow F_{selected} \cup \{f\}$
Train classifier C on F_{temp}
score ← Evaluate(C on validation set)
If score > best_score:
best_score ← score
best_feature ← f
Add best_feature to $F_{selected}$
End
Return $F_{selected}$

SBS Pseudocode for Classification Algorithm (Jain & Zongekar, 1997; Guyon & Elisseeff, 2003)

Input: Take D as the dataset, C as the classifier, and k as the number of features to select.

Output: Let selected features be $F_{selected}$
$F_{selected} \leftarrow$ All features
While |F_selected| > k:
best_score ← -∞
For each feature f in $F_{selected}$:
$F_{temp} \leftarrow F_{selected} \setminus \{f\}$
Train classifier C on F_{temp}
score ← Evaluate(C on validation set)
If score > best_score:
best_score ← score
worst_feature ← f
Remove worst_feature from $F_{selected}$
End
Return $F_{selected}$

GA Pseudocode for Classification Algorithm (Waheed et. al., 2020)

Input: Take D as the dataset, C as the classifier, and k as the number of features to select.

Output: Selected feature subset
Initialize population of N binary chromosomes (length = number of features)
For each generation from 1 to G:
For each chromosome in the population:
Select features where gene == 1
Train classifier C using selected features
Evaluate fitness using F1 score or Accuracy
Select top individuals via tournament/roulette
Perform crossover and mutation to create a new population
Return the chromosome with the highest fitness → selected feature subset

PSO Pseudocode for Classification Algorithm (Waheed et. al., 2020; Sivanandam & Deepa, 2007; Goldberg,1989)

Input: Take D as the dataset, C as the classifier, S as the swarm size, and T
Output: Best feature subset
Initialize S particles (binary vector positions & velocities)
For t = 1 to T:

For each particle:
Decode binary vector → selected features
Train classifier C with selected features
Compute fitness (e.g., F1 score)
Update personal best (pBest) and global best (gBest). For each particle:
Update velocity:
$v = \omega^* v + c_1{}^* r_1{}^*(\text{pBest - current}) + c_2{}^* r_2{}^*(\text{gBest - current})$
Update position using sigmoid(v) and threshold
Return gBest position (best feature subset)

DE Pseudocode for Classification Algorithm (Storn & Price, 1995; Das & Suganthan, 2011; Sharma & Kaur, 2020)

Input: Take D as the dataset, C as the classifier, P as the population size, and G as the generations.

Output: Best feature subset
Initialize population of binary vectors (length = number of features)
For generation = 1 to G:
For each target vector x in the population:
Randomly select $r_1, r_2, r_3 \neq x$
Mutation: $v = r_1 + F * (r_2 - r_3)$
Crossover: u = crossover(x, v, CR)
Binarize u with a threshold
If fitness(u) > fitness(x):
Replace x with u
Select the best individual in the population
Return as selected feature subset
The core mathematical functions are shown in the following table.

Algorithm	**Core Mathematical Function**
RFE	$S_{t+1}=S_t \setminus \arg\min I(f)$
SFS	$S_{t+1}=S_t \cup \arg\max J(S \cup f)$
SBS	$S_{t+1}=S_t \setminus \arg\max J(S - f)$
PSO	$v^{\{t+1\}} = \omega v^t + c_1 r_1(\text{pbest-x}) + c_2 r_2(\text{gbest-x})$
DE	$v = x_{r1} + F(x_{r2}-x_{r3})$
GA	$P(c_i) = J(c_i)/ \sum J$

Experimental Setup

For the purpose of analysis, the Python is used. Its high-level interactive nature and growing ecosystem of scientific libraries make it an attractive choice for algorithmic development and exploratory data analysis, as told by Dubois (2007) and Millman and Aivazis (2011). Nonetheless, its usage is expanding in both academic and industrial settings due to its general-purpose nature. Scikit-learn leverages this rich environment to provide state-of-the-art implementations of many well-known machine learning techniques while maintaining an easy-to-use interface that is closely related to the Python programming language.

Results and Analysis

Empirical studies have shown that wrapper methods can significantly improve the performance of machine learning models by selecting informative features that have a high predictive power. For example, in a study by Liu and Yu (2005), the authors applied wrapper methods for gene selection in cancer classification tasks and achieved better classification accuracy compared to filter-methods. Similarly, in a study by Saeys et al. (2007), the authors compared the performance of wrapper methods with filter methods in feature selection for microarray data analysis. They found that wrapper methods were more effective in selecting informative features for class prediction and outperformed filter methods in terms of model accuracy and generalization performance.

Ranks of the Features Using Six Wrapper Methods

The ranks of the features as identified by six wrapper methods are given in Table 1. It is evident from the ranks that there is a variation in ranks. Wrapper methods, as mentioned above, evaluate a feature subset based on the performance of the classification algorithms, which makes their ranking highly sensitive to search strategy, optimization goals, and interaction among the features. Deterministic methods such as RFE, SFS, and SBS often yield identical ranks since they rely on greedy, stepwise inclusion or exclusion of the features under fixed evaluation criteria. Meta-heuristic methods such as PSO, DE, and GA engage stochastic population-based searches that discover more broadly, capture non-linear relation dependencies, and higher-order interactions among features, leading to different ranks frequently.

Table 1:

Rank Comparison of Six Wrapper Methods

Feature	RFE Rank	SFS Rank	SBS Rank	PSO Rank	DE Rank	GA Rank
school	27	27	27	10	1	5
sex	11	11	11	3	4	6
age	14	14	14	12	14	8
address	16	16	16	6	11	19
famsize	22	22	22	1	5	7
Pstatus	28	28	28	20	20	24
Medu	5	5	5	30	28	10
Fedu	3	3	3	21	21	29
Mjob	1	1	1	28	13	16
Fjob	2	2	2	7	8	11
reason	4	4	4	14	10	1
guardian	6	6	6	4	12	14
traveltime	29	29	29	18	9	28
studytime	20	20	20	11	23	18
failures	7	7	7	5	26	27
schoolsup	21	21	21	8	18	17
famsup	19	19	19	25	24	12
paid	17	17	17	24	2	13
activities	12	12	12	17	19	26
nursery	23	23	23	26	3	4
higher	25	25	25	22	6	15
internet	24	24	24	15	7	25
romantic	15	15	15	23	29	23
famrel	18	18	18	19	17	22
freetime	13	13	13	9	30	20
goout	10	10	10	2	15	21
Dalc	30	30	30	27	16	2
Walc	26	26	26	16	22	30
health	8	8	8	13	27	9
absences	9	9	9	29	25	3

Wrapper methods are applied to various classification algorithms to check for the accuracy of models. Various algorithms like RF, SVM, KNN, LR, and DT are applied on the same dataset using a hybrid feature selection algorithm, resulting in similar results (Saxena & Singh, 2025).

Figure 1

Feature Importance Rank by using all Methods

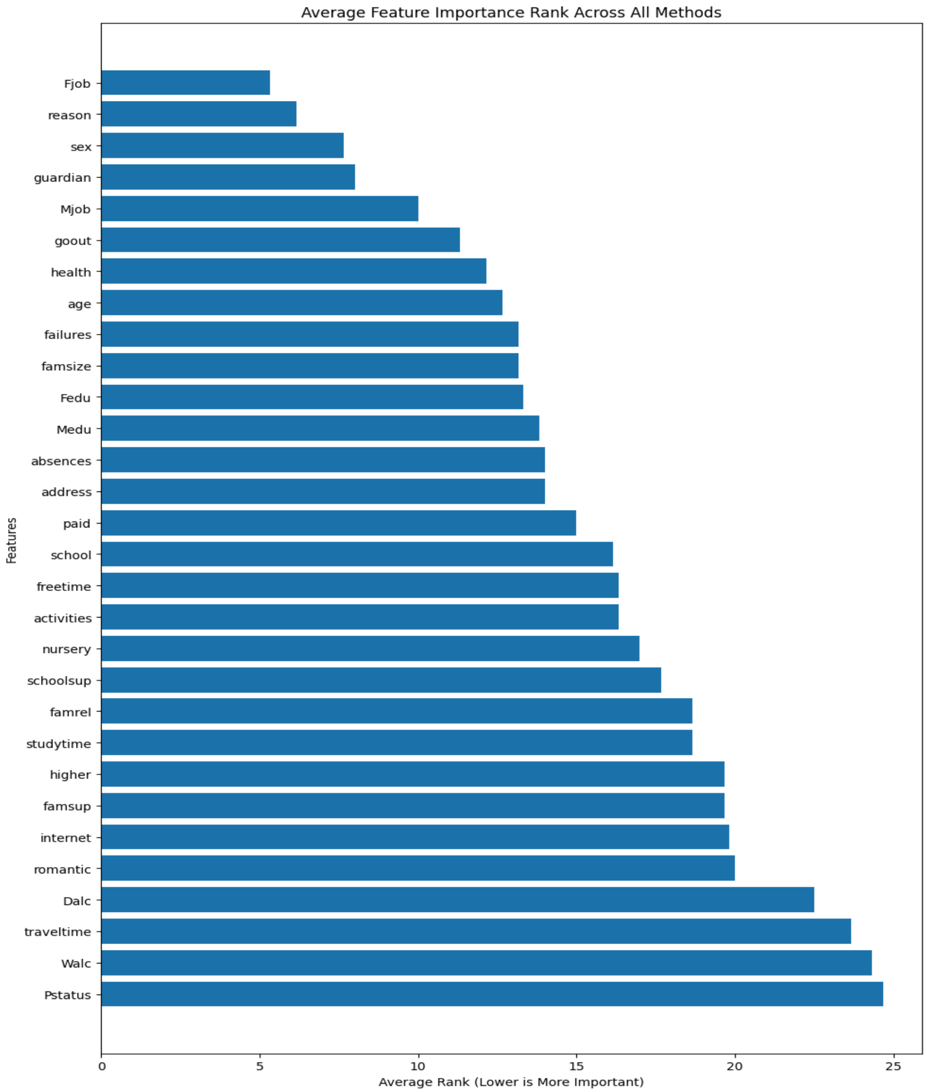


Figure 1 represents the rank of features by using Wrapper methods. It can be seen that the top features to select for the prediction of student academic performance are: Fjob, reason, guardian, Mjob, goout, paid, Medu, Fedu, and absences.

Performance of Classification Algorithms with all Features

The performance evaluation matrices of classification algorithms with all features of the data set are presented in Table 2. It can be inferred from the results given in table 2 that accuracy is highest for the combination of SVM (PSO); Precision is highest for the combination of Decision Tree (PSO); Recall is maximum for the combination of SVM (RFE) and SVM (SBS); F1-Score is highest for the combination SVM (RFE), SVM (SBS), and SVM (PSO); and AUC is highest for Random Forest (DE)

Table 2:

Performance Metrics with all Features using Classification Algorithm

Classification Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**AUC**
Decision Tree					
Decision Tree (RFE)	**0.64**	0.74	**0.71**	**0.73**	0.60
Decision Tree (SFS)	0.52	0.67	0.56	0.61	0.50
Decision Tree (SBS)	0.59	0.71	0.66	0.68	0.55
Decision Tree (PSO)	**0.64**	**0.75**	0.70	0.72	**0.61**
Decision Tree (GA)	0.59	0.69	**0.71**	0.70	0.52
Decision Tree (DE)	0.59	0.69	0.70	0.70	0.53
KNN					
KNN (RFE)	0.65	0.69	0.88	0.77	0.53
KNN (SFS)	0.61	0.67	0.84	0.74	0.48
KNN (SBS)	0.66	0.69	0.88	0.77	0.57
KNN (PSO)	0.61	0.66	0.84	0.74	0.48
KNN (GA)	**0.67**	**0.70**	**0.89**	**0.78**	0.60
KNN (DE)	0.64	0.69	0.84	0.76	**0.61**
Logistic Regression					
Logistic Regression (RFE)	0.66	**0.71**	0.84	0.77	0.58
Logistic Regression (SFS)	0.66	0.70	0.86	0.77	0.59
Logistic Regression (SBS)	0.64	0.69	0.83	0.75	0.58
Logistic Regression (PSO)	0.64	0.69	0.84	0.76	0.54

Classification Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**AUC**
Logistic Regression (GA)	**0.68**	**0.71**	0.90	**0.79**	**0.60**
Logistic Regression (DE)	0.65	0.69	0.85	0.76	0.59
Random Forest					
Random Forest (RFE)	0.66	0.70	0.86	0.77	0.55
Random Forest (SFS)	0.62	0.69	0.80	0.74	0.57
Random Forest (SBS)	**0.70**	**0.72**	**0.90**	**0.80**	0.54
Random Forest (PSO)	0.58	0.66	0.78	0.71	0.56
Random Forest (GA)	0.67	0.71	0.86	0.78	0.59
Random Forest (DE)	0.66	0.71	0.84	0.77	**0.62**
SVM					
SVM (RFE)	0.70	0.70	**0.96**	**0.81**	0.55
SVM (SFS)	0.67	0.69	0.93	0.79	0.57
SVM (SBS)	**0.71**	**0.71**	**0.96**	**0.81**	0.54
SVM (PSO)	0.70	**0.71**	0.94	**0.81**	0.57
SVM (GA)	0.67	0.70	0.91	0.79	0.60
SVM (DE)	0.67	0.69	0.93	0.79	**0.61**

Performance of Classification Algorithms With Top 20 Features

The performance evaluation matrices of classification algorithms with the top 20 features of the data set are presented in Table 3. It can be seen from the results that accuracy is highest for the combination of Logistic Regression (RFE); Precision is highest for the combination of Random Forest (SBS); Recall is maximum for the combination of SVM (RFE); F1-Score is highest for the combination of SVM (RFE); and AUC is highest for Random Forest (SBS). The performance of SVM classifiers in combination with RFE is best. It has an F1-score of 0.81 and a recall of 1.0.

Table 3:

Performance Metrics Calculation for Top 20 Features Using Classification Algorithm

Classification Model	Accuracy	Precision	Recall	F1 Score	AUC
Decision Tree					
Decision Tree (RFE)	0.61	**0.70**	0.74	0.72	0.54
Decision Tree (SFS)	0.53	0.67	0.60	0.63	0.49
Decision Tree (SBS)	**0.62**	**0.70**	**0.75**	**0.73**	0.55
Decision Tree (PSO)	**0.62**	**0.70**	**0.75**	**0.73**	0.55
Decision Tree (GA)	0.55	0.68	0.64	0.66	0.49
Decision Tree (DE)	0.61	0.73	0.66	0.69	**0.58**
KNN					
KNN (RFE)	**0.65**	0.69	0.85	**0.76**	0.52
KNN (SFS)	0.62	0.67	**0.87**	0.75	0.54
KNN (SBS)	0.59	0.66	0.81	0.73	**0.56**
KNN (PSO)	0.61	0.67	0.81	0.74	**0.56**
KNN (GA)	**0.65**	**0.70**	0.84	**0.76**	0.52
KNN (DE)	0.59	0.67	0.79	0.72	0.50
Logistic Regression					
Logistic Regression (RFE)	**0.70**	0.72	**0.91**	**0.80**	**0.62**
Logistic Regression (SFS)	0.62	0.67	0.85	0.75	0.60
Logistic Regression (SBS)	0.66	0.69	**0.91**	0.78	0.64
Logistic Regression (PSO)	0.66	0.69	0.89	0.78	0.59
Logistic Regression (GA)	0.69	0.71	0.90	**0.80**	0.57
Logistic Regression (DE)	0.66	0.69	**0.91**	0.78	0.57
Random Forest					
Random Forest (RFE)	0.65	0.69	0.87	0.77	0.57
Random Forest (SFS)	0.59	0.66	0.81	0.73	0.58
Random Forest (SBS)	0.68	0.72	0.87	**0.79**	**0.66**
Random Forest (PSO)	**0.68**	0.71	**0.89**	**0.79**	0.64
Random Forest (GA)	0.58	0.67	0.74	0.70	0.57
Random Forest (DE)	0.63	0.69	0.83	0.75	0.55
SVM					
SVM (RFE)	**0.68**	0.68	**1.00**	**0.81**	0.56
SVM (SFS)	0.65	0.67	0.92	0.78	0.61

Classification Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**AUC**
SVM (SBS)	0.65	0.67	0.92	0.78	0.61
SVM (PSO)	0.67	0.68	0.94	0.79	**0.62**
SVM (GA)	0.69	0.70	0.95	0.80	0.58
SVM (DE)	0.67	0.68	0.94	0.79	0.54

Performance of Classification Algorithms With Top 15 Features

The following inference can be drawn from the performance evaluation matrices of classification algorithms with the top 15 features presented in Table 4.

The combination with the highest accuracy is Random Forest (GA). The precision is highest for the combination of Decision Tree (SBS). The evaluation matrix recall is the maximum for the combination of SVM (RFE). F1-Score is highest for the combination SVM (RFE). AUC is highest for KNN (DE). The performance SVM (RFE) combination of classifiers and feature selection algorithm is best based on Recall (0.98) and F1-Score (0.81).

Table 4:

Performance Metrics for Top 15 Features Using Classification Algorithm

Classification Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**AUC**
Decision Tree					
Decision Tree (RFE)	0.52	0.64	0.66	0.65	0.45
Decision Tree (SFS)	0.54	0.69	0.58	0.63	0.46
Decision Tree (SBS)	0.66	**0.76**	**0.72**	**0.74**	0.62
Decision Tree (PSO)	0.56	0.70	0.60	0.65	0.53
Decision Tree (GA)	0.54	0.67	0.63	0.65	0.49
Decision Tree (DE)	**0.66**	0.76	**0.72**	0.74	0.63
KNN					
KNN (RFE)	0.62	0.66	**0.89**	0.76	0.54
KNN (SFS)	0.59	0.66	0.83	0.73	0.48
KNN (SBS)	0.63	0.68	0.87	0.76	0.59
KNN (PSO)	0.61	0.66	0.87	0.75	0.58
KNN (GA)	**0.64**	0.68	**0.89**	0.77	0.51
KNN (DE)	0.70	**0.72**	**0.89**	**0.80**	**0.69**

Classification Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**AUC**
Logistic Regression					
Logistic Regression (RFE)	**0.67**	**0.70**	0.89	**0.78**	0.57
Logistic Regression (SFS)	0.65	0.68	**0.91**	0.77	0.55
Logistic Regression (SBS)	0.66	0.69	**0.91**	**0.78**	**0.66**
Logistic Regression (PSO)	0.66	0.69	0.89	**0.78**	0.63
Logistic Regression (GA)	0.66	0.69	**0.91**	**0.78**	0.58
Logistic Regression (DE)	**0.67**	**0.70**	0.89	**0.78**	0.65
Random Forest					
Random Forest (RFE)	0.59	0.66	0.83	0.73	0.55
Random Forest (SFS)	0.53	0.65	0.66	0.65	0.53
Random Forest (SBS)	0.70	**0.75**	0.83	0.79	0.60
Random Forest (PSO)	0.67	0.73	0.81	0.77	**0.62**
Random Forest (GA)	**0.71**	0.73	**0.89**	**0.80**	0.58
Random Forest (DE)	0.68	0.72	0.87	0.79	0.70
SVM					
SVM (RFE)	0.70	0.69	**0.98**	**0.81**	0.54
SVM (SFS)	0.66	0.68	0.92	0.78	0.51
SVM (SBS)	0.65	0.68	0.91	0.77	**0.62**
SVM (PSO)	0.65	0.68	0.91	0.77	0.60
SVM (GA)	0.69	**0.70**	0.95	0.80	0.56
SVM (DE)	0.68	0.69	0.96	0.80	0.63

Performance of Classification Algorithms With Top 10 Features

The following inference can be drawn from the performance evaluation matrices of classification algorithms with the top 10 features presented in Table 5.

Accuracy is highest for the combination of Random Forest (SBS). Precision is highest for the combination of Random Forest (SBS). The value of recall is maximum for the combination of SVM (RFE). F1-Score is highest for the combination SVM (RFE). AUC is highest for KNN (GA). The performance of SVM (RFE) is better on all evaluation parameters except AUC. Similarly performance of the combination Decision Tree (SFS) is good for all the parameters except AUC.

Table 5:

Performance Metrics for Top 10 Features Using Classification Algorithm

Classification Model	Accuracy	Precision	Recall	F1 Score	AUC
Decision Tree					
Decision Tree (RFE)	0.49	0.64	0.57	0.60	0.46
Decision Tree (SFS)	**0.65**	0.68	**0.89**	**0.77**	0.47
Decision Tree (SBS)	0.61	**0.72**	0.68	0.70	**0.57**
Decision Tree (PSO)	0.61	**0.72**	0.68	0.70	**0.57**
Decision Tree (GA)	0.57	**0.72**	0.58	0.65	**0.57**
Decision Tree (DE)	0.58	0.70	0.66	0.68	0.54
KNN					
KNN (RFE)	0.62	0.68	0.81	0.74	0.50
KNN (SFS)	0.58	0.65	0.83	0.73	0.45
KNN (SBS)	0.68	**0.73**	0.85	0.78	0.63
KNN (PSO)	0.62	0.67	0.85	0.75	0.55
KNN (GA)	0.66	0.69	**0.91**	0.78	**0.71**
KNN (DE)	**0.70**	**0.73**	0.87	**0.79**	0.60
Logistic Regression					
Logistic Regression (RFE)	0.66	0.68	**0.92**	**0.78**	0.57
Logistic Regression (SFS)	0.66	0.68	**0.92**	**0.78**	0.57
Logistic Regression (SBS)	0.65	0.68	0.91	0.77	**0.67**
Logistic Regression (PSO)	0.63	0.67	0.89	0.76	0.63
Logistic Regression (GA)	0.65	0.69	0.87	0.77	0.61
Logistic Regression (DE)	**0.67**	**0.70**	0.89	**0.78**	0.66
Random Forest					
Random Forest (RFE)	0.59	0.67	0.77	0.72	0.53
Random Forest (SFS)	0.63	0.68	**0.87**	0.76	0.54
Random Forest (SBS)	**0.71**	**0.77**	0.81	**0.79**	**0.61**
Random Forest (PSO)	0.61	0.68	0.79	0.73	0.59
Random Forest (GA)	0.62	0.69	0.79	0.74	0.64
Random Forest (DE)	0.66	0.70	0.87	0.77	0.60
SVM					
SVM (RFE)	**0.68**	**0.69**	**0.96**	**0.80**	0.54
SVM (SFS)	0.66	0.68	0.92	0.78	0.47

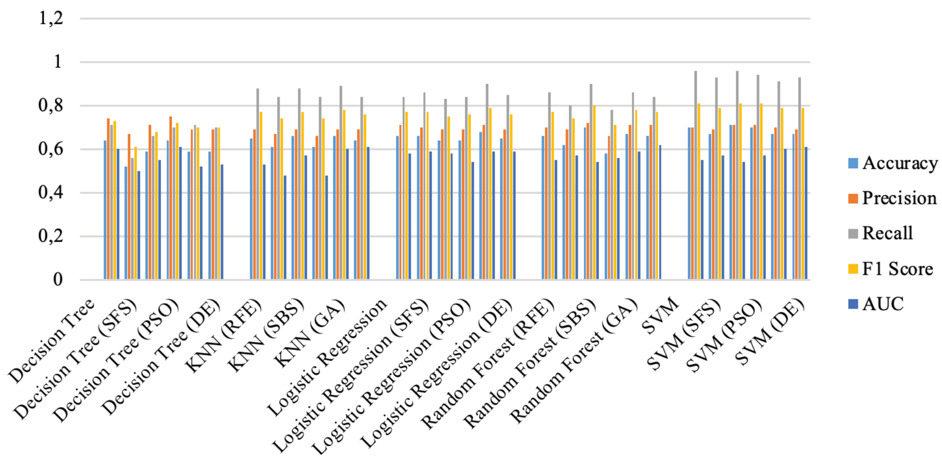
Classification Model	Accuracy	Precision	Recall	F1 Score	AUC
SVM (SBS)	0.67	**0.69**	0.92	0.79	0.58
SVM (PSO)	0.67	0.68	0.94	0.79	**0.61**
SVM (GA)	0.67	0.68	0.94	0.79	0.58
SVM (DE)	0.66	0.68	0.92	0.78	**0.61**

Comparative Analysis

The graphical visualization of the performance evaluation matrices of five classification algorithms in conjunction with wrapper algorithms is given in Figures 2 to 5.

Figure 2:

Performance Metrics of Classification Models based on All Features



In Figure 2, performance metrics are used to represent various classification algorithms. From the above, it can be seen that SVM performs better than other classification algorithms as per the values of recall, precision, and accuracy. In Figure 3, performance metrics are given for five classification algorithms in conjunction with six feature selection methods for the top 20 features. It can be seen from the bar charts that SVM performs better than other classification algorithms in combination with six algorithms based on the Recall values. For the remaining combination, there is no specific trend. However, the combination of Decision Tree (SBS) and Decision Tree (PSO) performs well based on all parameters except AUC.

Figure 3:

Performance Metrics of Classification Models based on Top 20 Features

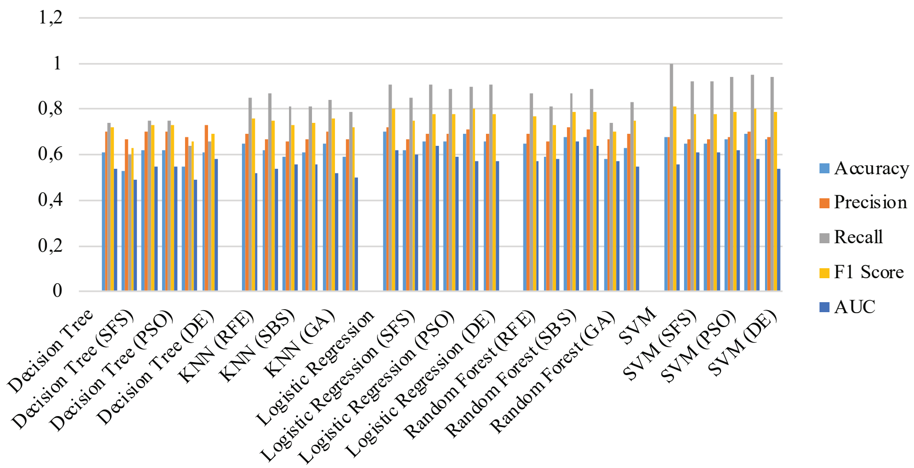


Figure 4:

Performance Metrics of Classification Models based on Top 15 Features

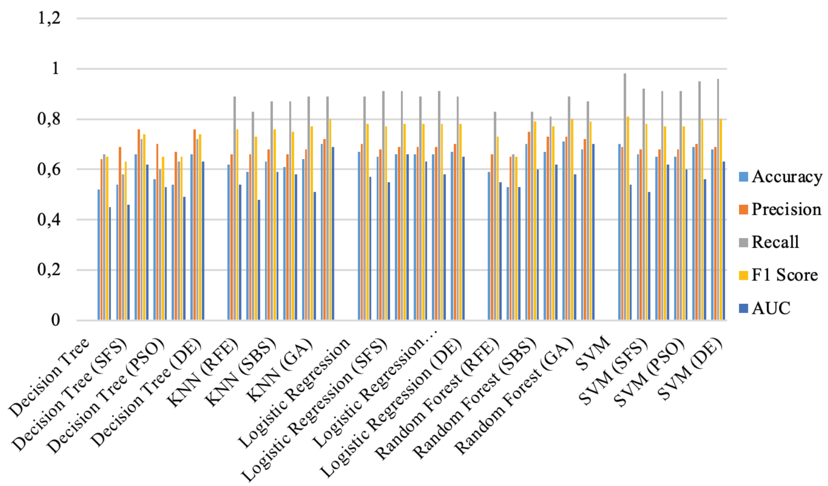


Figure 4 depicts the performance metrics of five classification algorithms in combination with wrapper methods. From the figure 4, it can be seen that SVM performs better than other classification algorithms based on recall. This graph is used to represent the Top 15 features.

Figure 5:

Performance Metrics of Classification Models based on Top 10 Features

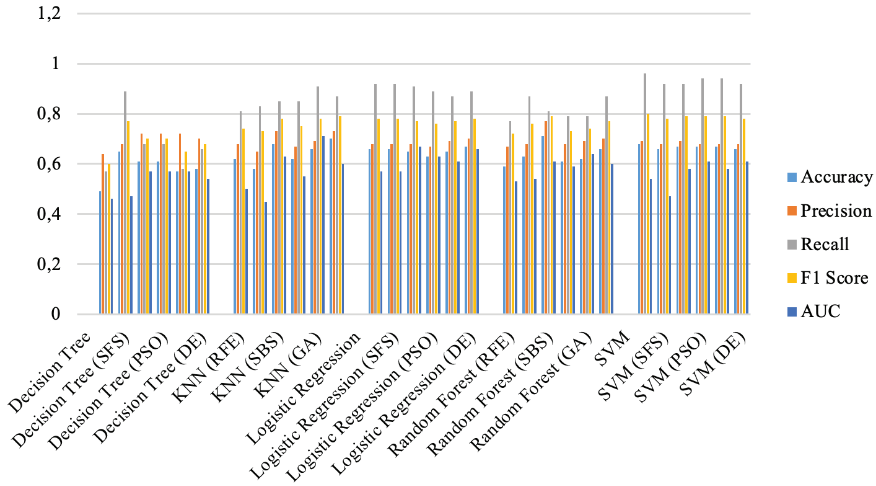


Figure 5 depicts the performance metrics for the five classification algorithms in conjunction with six wrapper algorithms. It is evident from Figure 5 that SVM performs better than other classification algorithms in conjunction with all six wrapper methods based on the recall. This graph is used to represent the Top 10 features.

Conclusion

It has been seen that from the data presented in Table 2-5 and Figures 2- 5 that the optimal combination of feature selection and classification algorithm varies with the evaluation matrix and number of features selected for fitting the classification algorithm. The summary of the best combinations is given in Table 6. To be specific, when all features are considered, SVM (PSO) is best based on Accuracy, and the decision tree (PSO) on the basis of precision. The SVM (RFE) and SVM (SBS) combination is best on recall and F1-score, demonstrating greater sensitivity in recognizing successful students. On the other hand, the highest AUC is observed for the combination random forest (DE), suggesting that this combination has better discriminating capabilities. Similar conclusions can be drawn for other combinations also.

In summary, these results indicate that SVM combined with RFE is the most consistent combination, particularly based on recall and F1-score. It shows that deterministic, greedy wrapper feature selection algorithms such as RFE are effective in combination with a margin-based classifier, i.e., SVM. On the other hand, evolutionary swarm-based wrapper (GA and DE) shows better strength in combination with ensemble or instance-based classifiers, i.e., random forest and KNN, based on AUC and accuracy.

Table 6:

Best Feature Selection and Classifier Combinations based on Evaluation Metrics

Feature Set Size	Metric	Best Combination (Wrapper + Classifier)	Value
All Features	Accuracy	SVM (PSO)	0.70
	Precision	Decision Tree (PSO)	0.75
	Recall	SVM (RFE) / SVM(SBS)	0.96
	F1-score	SVM (RFE) / SVM (SBS) / SVM (PSO)	0.81
	AUC	Random Forest (DE)	0.62
Top 20 Features	Accuracy	Logistic Regression (RFE)	0.70
	Precision	Random Forest (SBS)	0.72
	Recall	SVM (RFE)	1.00
	F1-score	SVM (RFE)	0.81
	AUC	Random Forest (SBS)	0.66
Top 15 Features	Accuracy	Random Forest (GA)	0.71
	Precision	Decision Tree (SBS / DE)	0.76
	Recall	SVM (RFE)	0.98
	F1-score	SVM (RFE)	0.81
	AUC	KNN + DE	0.69
Top 10 Features	Accuracy	Random Forest (SBS)	0.71
	Precision	Random Forest (SBS)	0.77
	Recall	SVM (RFE)	0.96
	F1-score	SVM (RFE)	0.80
	AUC	KNN (GA)	0.71

Note. The **Best Overall Combination** is determined based on consistency across feature subset sizes and balanced performance on recall and F1-score, which are critical metrics for educational outcome prediction.

In this research paper, the impact of Wrapper methods is studied with the help of a datasets and their performance is analyzed based on rank and quality parameters such as precision, accuracy, F1 Score, Recall, and AUC Curve. Based on the results presented in the earlier sections, the following can be inferred.

It is also observed that reducing the feature set to the top 10, 15, or 20 features did not significantly degrade model performance. In many cases, using fewer features led to even higher accuracy and generalization, supporting the practical utility of wrapper methods in reducing dimensionality without sacrificing predictive power. Future work can focus on combining wrapper methods with filter-based techniques to create hybrid approaches that are both computationally efficient and highly accurate. There is also scope for extending the framework to multiclass classification tasks and applying deep learning models in conjunction with feature selection.

References

Asif, R., Merceron, A., & Pathan, M. K. (2017). Predicting student academic performance at degree level: A case study. *International Journal of Intelligent Systems and Applications, 9*(12), 49–61. https://doi.org/10.5815/ijisa.2017.12.06

Bin Mat, U., Buniyamin, N., Arsad, P. M., & Kassim, R. A. (2014). An overview of using academic analytics to predict and improve students' achievement: A proposed proactive intelligent intervention. In *Proceedings of the 2013 IEEE 5th International Conference on Engineering Education (ICEED)* (pp. 126–130). IEEE.

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering, 40*(1), 16–28.

Das, S., & Suganthan, P. N. (2011). Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation, 15*(1), 4–31. https://doi.org/10.1109/TEVC.2010.2059031

Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis, 1*(3), 131–156.

Dubois, P. F. (2007). Python: Batteries included. *Computing in Science & Engineering, 9*(3), 7–9. https://doi.org/10.1109/MCSE.2007.58

Feng, C., & Xu, X. (2019). A hybrid feature selection method based on RFE and Relief for educational data mining. *Computers in Human Behavior, 101*, 228–237. https://doi.org/10.1016/j.chb.2019.07.002

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research, 3*, 1157–1182.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning, 46*(1–3), 389–422. https://doi.org/10.1023/A:1012487302797

Holland, J. H. (1992). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press.

Islam, M. M., & Murase, K. (2018). A hybrid wrapper–filter approach for feature selection using mutual information and differential evolution. *Knowledge-Based Systems, 147*, 125–134. https://doi.org/10.1016/j.knosys.2018.01.002

Jain, A. K., & Zongker, D. (1997). Feature selection: Evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(2), 153–158. https://doi.org/10.1109/34.574797

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks* (pp. 1942–1948). IEEE.

Khan, M. J., & Jawaid, A. (2020). Prediction of at-risk students using genetic algorithm and random forest. *Education and Information Technologies, 25*(4), 2979–2997.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence, 97*(1–2), 273–324.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature selection: A data perspective. *ACM Computing Surveys, 50*(6), 1–45.

Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering, 17*(4), 491–502. https://doi.org/10.1109/TKDE.2005.66

Maldonado, S., & Weber, R. (2009). A wrapper method for feature selection using support vector machines. *Information Sciences, 179*(13), 2208–2217.

Millman, K. J., & Aivazis, M. A. (2011). Python for scientists and engineers. *Computing in Science & Engineering, 13*(2), 9–12. https://doi.org/10.1109/MCSE.2011.36

Moslehi, F., & Haeri, A. (2020). A novel hybrid wrapper–filter approach based on genetic algorithm and particle swarm optimization for feature subset selection. *Journal of Ambient Intelligence and Humanized Computing, 11*, 1105–1127.

Patel, D., Saxena, A., & Wang, J. (2024). A machine-learning-based wrapper method for feature selection. *International Journal of Data Warehousing and Mining*. https://doi.org/10.4018/IJDWM.352041

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

Pudil, P., Novovicová, J., & Kittler, J. (1994). Floating search methods in feature selection. *Pattern Recognition Letters, 15*(11), 1119–1125. https://doi.org/10.1016/0167-8655(94)90127-9

Saeys, Y., Inza, I., & Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics, 23*(19), 2507–2517.

Saxena, M., & Singh, N. P. (2025). Role of Hybrid Feature Selection Algorithms in Foreseeing Stu dent Performance. International Journal of Natural and Technical Sciences, 5(1), 43-75. https://doi.org/10.69648/NVAX8158

Sharma, M., & Kaur, P. (2020). An efficient differential evolution approach for feature selection in educational data mining. *International Journal of Cognitive Computing in Engineering, 1*, 33–43.

Siedlecki, W., & Sklansky, J. (1989). A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters, 10*(5), 335–347.

Singh, A. K., & Karthikeyan, S. (2024). A wrapper feature selection method for predicting student dropout in higher education. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.5002077

Sivanandam, S. N., & Deepa, S. N. (2007). *Introduction to genetic algorithms*. Springer. https://doi.org/10.1007/978-3-540-73190-0

Storn, R., & Price, K. (1997). Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*(4), 341–359.

Tang, J., Alelyani, S., & Liu, H. (2014). Feature selection for classification: A review. In H. Liu & L. Yu (Eds.), *Data classification: Algorithms and applications* (pp. 37–64). CRC Press.

Tomasevic, N., Gvozdenovic, N., & Vranes, S. (2019). An overview and comparison of supervised data mining techniques for student exam performance prediction. *Computers & Education, 143*, 103676. https://doi.org/10.1016/j.compedu.2019.103676

Waheed, H., Goyal, M., & Khanum, A. (2020). Machine learning-based early prediction of students' performance using PSO. *Education and Information Technologies, 25*(6), 4693–4711.

Xue, B., Zhang, M., & Browne, W. N. (2015). A comprehensive comparison on evolutionary feature selection approaches to classification. *International Journal of Computational Intelligence and Applications, 14*(2), 1550008.

Xue, B., Zhang, M., & Browne, W. N. (2016). A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation, 20*(4), 606–626.