Predicting Factors Influencing Student Academic Performance

Submitted by

Joyston Jose D’souza

24251325

MSc. Big Data Analytics

AIMIT, St. Aloysius (Deemed to be University)

Mangalore, Karnataka

Submitted in Partial Fulfillment of the Requirements for the Award of the Degree of

Master of Big Data Analytics

Under the guidance of

Dr. Hemalatha N
Dean, School of IT
AIMIT, St. Aloysius (Deemed to be University),
Mangaluru-575 022.

Submitted to

ALOYSIUS INSTITUTE OF MANAGEMENT AND INFORMATION TECHNOLOGY
(AIMIT)
ST ALOYSIUS COLLEGE (DEEMED TO BE UNIVERSITY)
MANGALORE, KARNATAKA

2025

ABSRACT

This study addresses the problem of identifying high-performing students using behavioural and demographic survey data. Machine learning has become an essential component of educational data mining, facilitating exploration of complex learning factors and enabling early student support [1]. Prior work highlights that such predictive models inform learning analytics frameworks and aid early identification of at-risk students [1]. Data preprocessing involved SMOTE for class balancing and feature standardization. Four classifiers—Logistic Regression, Random Forest, Support Vector Machine, and Multilayer Perceptron—were evaluated. Random Forest outperformed others, achieving 74% test accuracy and 87% accuracy under 10-fold cross-validation. These results indicate that nonlinear models capture complex educational patterns, underscoring the potential of ML-driven predictive analyst [1].

INDEX

1. INTRODUCTION

Predicting student academic performance is a major goal in educational data mining and learning analytics. Early identification of students who may struggle allows educators to intervene and improve outcomes [1]. Data-driven predictive models can reveal which factors (study habits, attendance, sleep, etc.) correlate with success. This project addresses the binary classification task of predicting whether a student is a "high performer" based on survey responses about their study and lifestyle habits. Such analysis can inform tailored support strategies and contribute to a data-informed educational framework [1].

In this study, we use a provided dataset of student features (e.g., "Studies Daily", "Attends Extra Classes", "Sleeps at Least 7 Hours", etc., all recorded as Yes/No) and a target label "High Performer" (Yes/No). We implement and compare four machine learning algorithms: Logistic Regression, Random Forest, Support Vector Machine, and a multilayer neural network (MLP). Each classifier's motivation is briefly as follows:

- Logistic Regression: A simple linear model for binary classification. It estimates the probability of being a high performer under a logistic model [2].
- Random Forest: An ensemble of decision trees (Breiman, 2001) that reduces variance through bagging. It can capture nonlinear interactions and is robust to feature types [3].
- Support Vector Machine (SVM): A classifier that finds a maximum-margin hyperplane (potentially in a transformed feature space via kernels). Good for complex but lower-dimensional data [4].
- Multilayer Perceptron (MLP): A feedforward neural network with hidden layers. It can model complex nonlinear relationships given sufficient data [5].

Each model was trained on 80% of the dataset and evaluated on the remaining 20% as a held-out test set. To further assess model robustness and generalization, we employed both 5-fold and 10-fold cross-validation techniques [6]. These validation strategies split the training data into multiple subsets, iteratively training on a portion and validating on the remainder, thereby reducing the likelihood of overfitting and ensuring a more reliable estimate of performance [6]. Evaluation metrics included confusion matrices and classification metrics such as precision, recall, F1-score (for each class), macro and weighted averages, and overall accuracy [7]. Cross-validation results were particularly insightful: Random Forest achieved the highest 10-fold cross-validation accuracy (87%), indicating strong generalization, while MLP closely followed with 86%. These combined evaluation strategies provide a comprehensive view of binary classification performance and highlight the relative consistency of each model.

2. MATERIAL AND METHOD

2.1. Dataset Description

The dataset contains 310 student records with 22 columns: 21 predictor attributes and 1 target label. Each attribute is categorical (mostly Yes/No) or binary (e.g. "Gender" as Female/Male, "School Type" as Private/Public). A brief listing of the features is as follows:

- Gender, School Type: Demographics (e.g. Male/Female, Public/Private).
- Studies Daily, Attends Extra Classes, Uses Online Learning Platforms, Has a Fixed Study Schedule, Participates in Group Study, Sleeps at Least 7 Hours, Uses Social Media During Study Hours, Submits Assignments on Time, Enjoys Reading, Participates in School Activities, Uses a Tutor, Prefers Studying Alone, Attends School Regularly, Has a Part-time Job, Uses a Planner for Schoolwork, Gets Nervous Before Exams, Prefers Online Classes Over In-person, Takes Notes in Class, Has a Quiet Study Environment at Home: Behavioural or attitudinal factors (Yes/No).
- High Performer: Target label (Yes=1 means high performance, No=0 otherwise). In total, the predictors represent attendance and study habits hypothesized to influence outcomes [1]. The data are entirely categorical (Yes/No or binary). There were no missing values in any column. The target distribution was somewhat imbalanced: 180 of 310 students (58%) were labelled high performers (Yes) and 130 (42%) low performers (No). Table 1 below summarizes the data dimensions:

(*Table 1: Dataset summary and target class distribution.*)

	Count
Total rows	310
Total cols	22
High Performer = Yes	180
High Performer = No	130

2.2. Data Preprocessing

Data preprocessing is a critical step in any machine learning project. It ensures that raw data is transformed into a clean and structured format suitable for modelling [8]. In this study, preprocessing involved handling missing values, encoding categorical features into numerical formats, balancing the class distribution using SMOTE, and standardizing feature values. These transformations were essential to improve model performance and ensure fair evaluation across all algorithms [8].

2.2.1. Missing Values

There were no missing values present in the dataset. All records were complete, and each feature column contained valid values, removing the need for any imputation or data cleaning.

2.2.2. Encoding

All categorical features such as 'Yes/No' or 'Male/Female' were encoded numerically using Label Encoding [9]. Each category was converted into binary format (0/1) using Scikit-learn's Label Encoder, enabling compatibility with machine learning algorithms [9].

2.2.3. Skewness

In binary categorical data such as Yes/No responses, skewness refers to the imbalance between the frequency of “Yes” and “No”. A highly skewed feature may introduce bias into the learning process if one class dominates. To quantify this, we used the normalized absolute difference:

$$\mathrm{Skewness} = \frac{|\mathrm{N_{Yes}} - \mathrm{N_{No}}|}{\mathrm{N_{Yes}} + \mathrm{N_{No}}}$$

This measure ranges from 0 (perfect balance) to 1 (complete imbalance) [10]. Skewness was computed across all binary behavioural features. Features like:

- Prefers Online Classes Over In-person
- Uses Online Learning Platforms
- Takes Notes in Class

were found to be the most skewed. This information guides our understanding of how evenly distributed each attribute.

2.2.4. SMOTE

The original dataset had a class imbalance: 58% of students were labelled as 'High Performer'. To address this, SMOTE (Synthetic Minority Over-sampling Technique) was used to balance the training set [11]. It synthetically generates new examples from the minority class, ensuring equal class distribution during training [11].

2.2.5. Standardization

Features were standardized using Scikit-learn's StandardScaler [12]. This transformation set all features to zero mean and unit variance. Standardization is especially important for algorithms such as Support Vector Machines and Neural Networks, which are sensitive to the scale of input features [12].

2.3. Algorithm

Various machine learning algorithms were employed to build classification models capable of predicting whether a student is a high academic performer. Each algorithm brings unique strengths and assumptions [2][3][4][5]. This study compares both linear (Logistic Regression) and non-linear (Random Forest, SVM, MLP) classifiers to evaluate their effectiveness on the processed dataset. These algorithms were selected for their popularity, interpretability, and ability to capture complex patterns in the data.

2.3.1. Logistic Regression

Logistic Regression is a linear classifier that estimates the probability of class membership using a logistic function [2]. It was used as a baseline model. However, due to its linear nature, it showed limited accuracy (61%) on this dataset.

2.3.2. Random Forest Classification

Random Forest is an ensemble of decision trees built using the bagging method [3]. It is effective for modelling nonlinear patterns and handling categorical data. This model achieved 74% accuracy, outperforming the logistic regression.

2.3.3. Support Vector Machine (SVM)

SVM seeks to find the optimal hyperplane that maximizes the margin between two classes. A linear kernel was used in this project [4]. After standardization, SVM achieved a high classification accuracy of 76%.

2.3.4. Multilayer Perceptron (MLP)

MLP is a deep learning model consisting of multiple hidden layers [5]. A configuration of three hidden layers, each with 100 neurons, was used. It matched SVM performance with an accuracy of 76%, demonstrating its capability to model complex nonlinear relationships.

2.4. Performance Matrix

To objectively evaluate the performance of each model, several classification metrics were used. These include the confusion matrix, precision, recall, F1-score, and overall accuracy [7]. Metrics such as macro and weighted averages offer a more balanced view, especially in imbalanced datasets. Additionally, cross-validation techniques like 5-fold and 10-fold CV were applied to ensure the robustness and generalization capability of the models across different data splits [6].

2.4.1. Confusion Matrix

A confusion matrix shows the number of correct and incorrect predictions made by the classifier. It helps identify false positives, false negatives, true positives, and true negatives, giving insight into model performance [7].

2.4.2. Classification Report

Model performance was evaluated using classification metrics: confusion matrix, precision, recall, F1-score, and accuracy. Macro and weighted averages were included to address class imbalance. Robustness and generalization were ensured through 5-fold and 10-fold cross-validation [6][7].

2.4.2.1. Precision

Precision is the ratio of true positive predictions to the total predicted positives. It reflects the exactness of the model [7].

$$\text{Precision} = \frac{TP}{TP + FP}$$

2.4.2.2. F1 Score

The F1 score is the harmonic mean of precision and recall. It is useful for evaluating model performance on imbalanced datasets [7].

$$F_1 = \frac{2 \times (Precision \times Recall)}{Precision + Recall}$$

2.4.2.3. Support

Support refers to the actual number of instances for each class. It helps understand the impact of class distribution on metrics [7].

2.4.2.4. Accuracy

Accuracy is the ratio of correct predictions to the total number of predictions. It gives a general measure of classifier effectiveness [7].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

2.4.2.5. Macro Average

Macro average calculates metrics independently for each class and takes the unweighted mean, treating all classes equally [7].

2.4.2.6. Weighted Average

Weighted average considers the number of true instances for each class and computes a weighted mean accordingly. It accounts for class imbalance in performance measurement [7].

2.4.3. Cross-validation

To assess generalization, 5-fold and 10-fold cross-validation techniques were used. These methods split the dataset into multiple parts for training and testing, reducing the risk of overfitting and ensuring stable evaluation metrics [6].

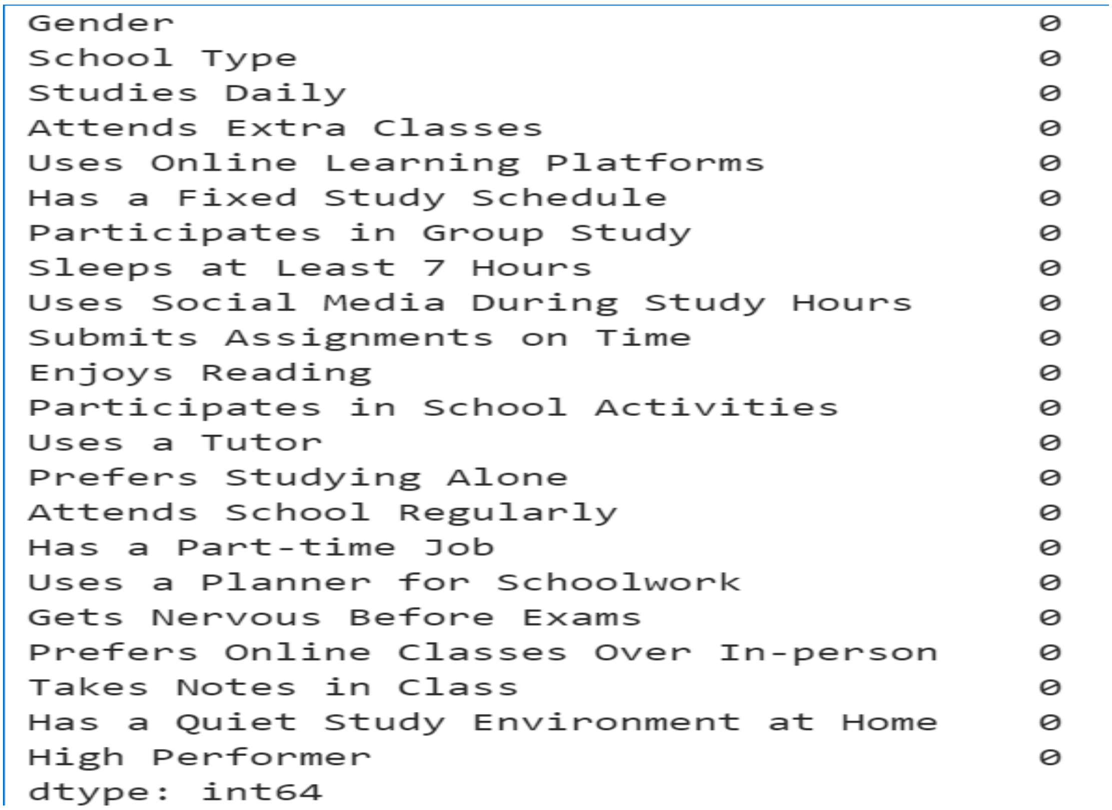
3. Result

3.1. Data Preprocessing Result

3.1.1. Missing Values

No missing values were present in the dataset.

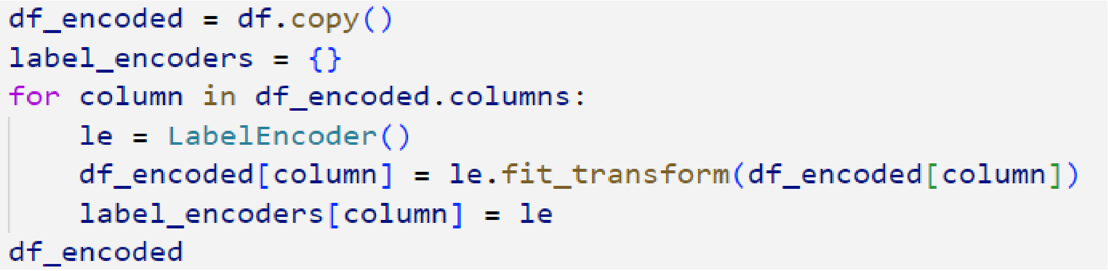
(Table 2: Missing Values Count)



```
Gender                                        0
School Type                                   0
Studies Daily                                 0
Attends Extra Classes                         0
Uses Online Learning Platforms                0
Has a Fixed Study Schedule                    0
Participates in Group Study                   0
Sleeps at Least 7 Hours                       0
Uses Social Media During Study Hours          0
Submits Assignments on Time                   0
Enjoys Reading                                0
Participates in School Activities             0
Uses a Tutor                                  0
Prefers Studying Alone                        0
Attends School Regularly                      0
Has a Part-time Job                           0
Uses a Planner for Schoolwork                 0
Gets Nervous Before Exams                     0
Prefers Online Classes Over In-person         0
Takes Notes in Class                          0
Has a Quiet Study Environment at Home         0
High Performer                                0
dtype: int64
```

3.1.2. Encoding

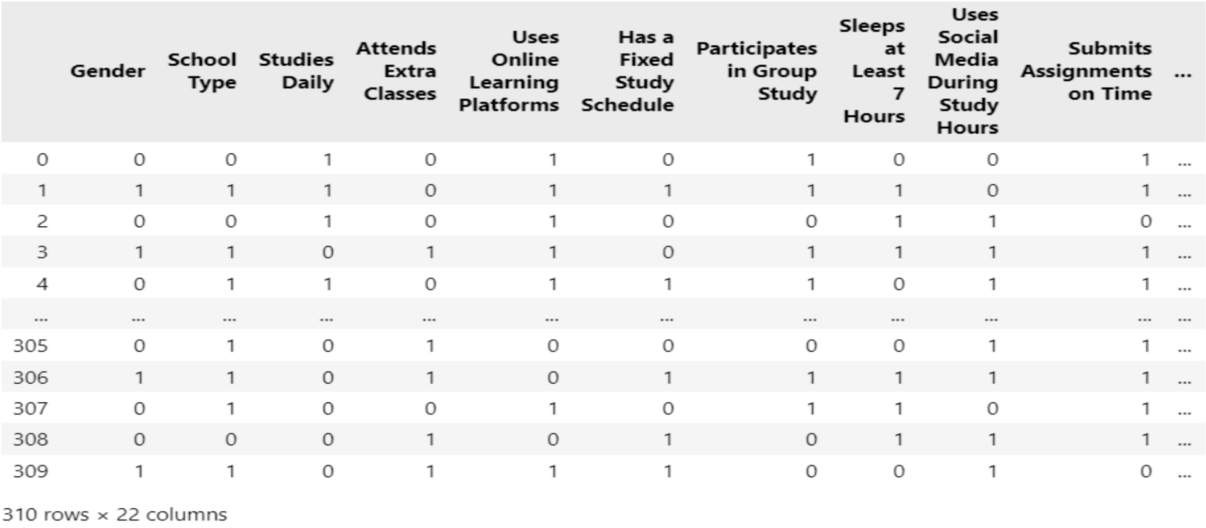
All categorical variables (Yes/No, Male/Female) were label encoded to binary (0/1) [9].



```
df_encoded = df.copy()
label_encoders = {}
for column in df_encoded.columns:
    le = LabelEncoder()
    df_encoded[column] = le.fit_transform(df_encoded[column])
    label_encoders[column] = le
df_encoded
```

(Figure 1: Snippet code of Encoding)

(Table 2.1: Encoding Yes/No as 1/0)

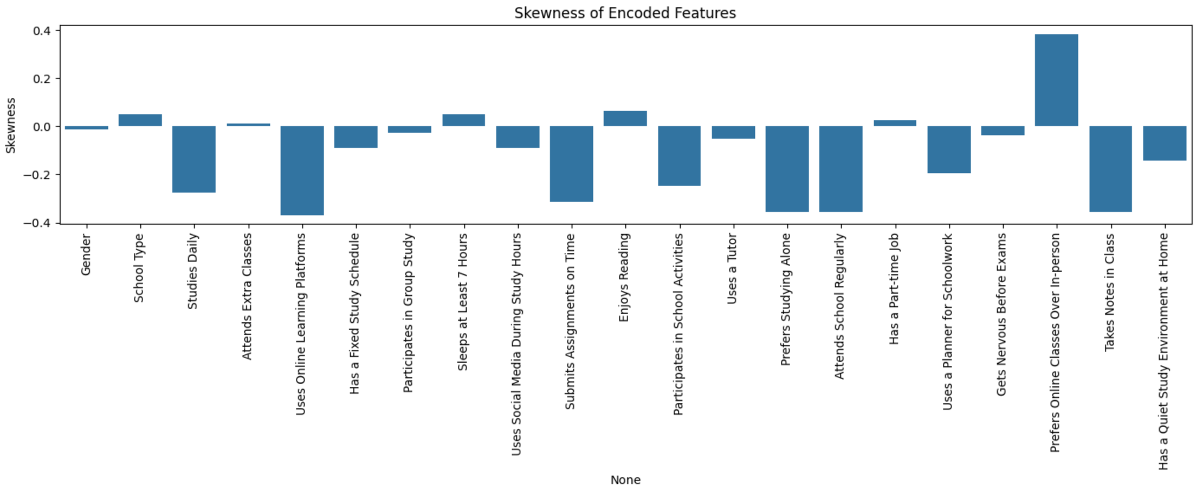


	Gender	School Type	Studies Daily	Attends Extra Classes	Uses Online Learning Platforms	Has a Fixed Study Schedule	Participates in Group Study	Sleeps at Least 7 Hours	Uses Social Media During Study Hours	Submits Assignments on Time	...
0	0	0	1	0	1	0	1	0	0	1	...
1	1	1	1	0	1	1	1	1	0	1	...
2	0	0	1	0	1	0	0	1	1	0	...
3	1	1	0	1	1	0	1	1	1	1	...
4	0	1	1	0	1	1	1	0	1	1	...
...	...	...	...	...	...	...	...	...	...	...	...
305	0	1	0	1	0	0	0	0	1	1	...
306	1	1	0	1	0	1	1	1	1	1	...
307	0	1	0	0	1	0	1	1	0	1	...
308	0	0	0	1	0	1	0	1	1	1	...
309	1	1	0	1	1	1	0	0	1	0	...

310 rows × 22 columns

3.1.3. Skewness

Most features were balanced; a few showed high skewness [10].

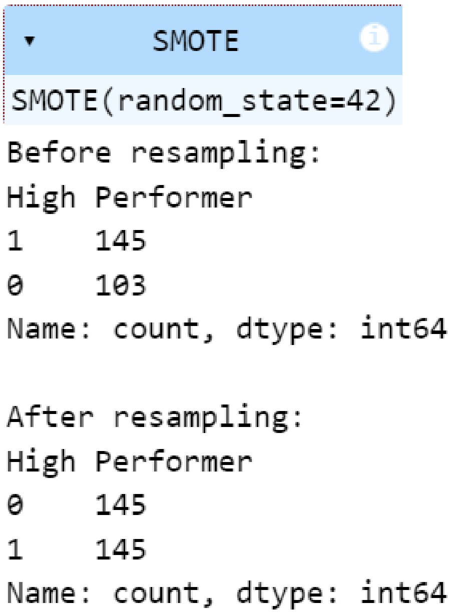


(Figure 1.1: Feature Skewness)

3.1.4. SMOTE

SMOTE was applied to balance the dataset. After SMOTE, both classes had 145 instances, improving model training [11].

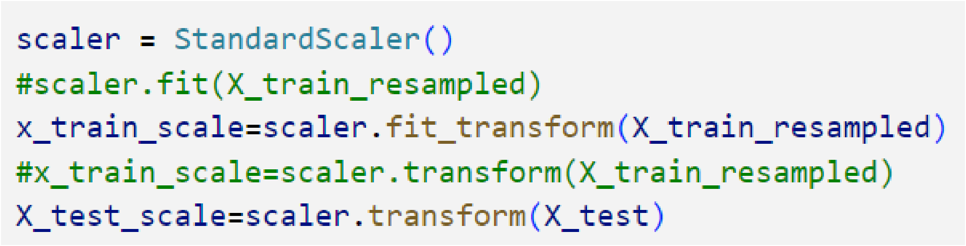
(Table 2.2: SMOTE)



```
▾        SMOTE
SMOTE(random_state=42)
Before resampling:
High Performer
1    145
0    103
Name: count, dtype: int64

After resampling:
High Performer
0    145
1    145
Name: count, dtype: int64
```

3.1.5. Standardization

All features were standardized (mean 0, variance 1) for optimal model performance [12].



```
scaler = StandardScaler()
#scaler.fit(X_train_resampled)
x_train_scale=scaler.fit_transform(X_train_resampled)
#x_train_scale=scaler.transform(X_train_resampled)
X_test_scale=scaler.transform(X_test)
```

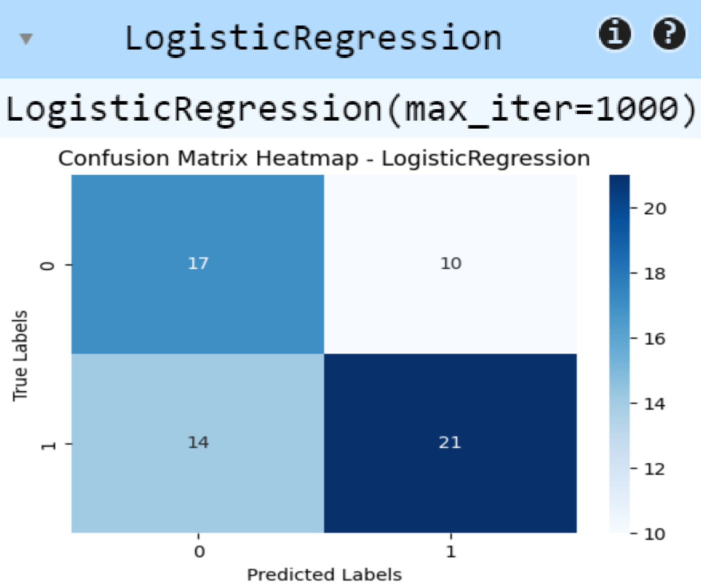
(Figure 1.2 : Snippet code of standardization)

3.2. Algorithm Result

All models were trained on 80% of the data and tested on the remaining 20%. The following sections summarize the results with performance metrics and visualizations (confusion matrices, classification reports, and accuracy) [7].

3.2.1. Logistic Regression

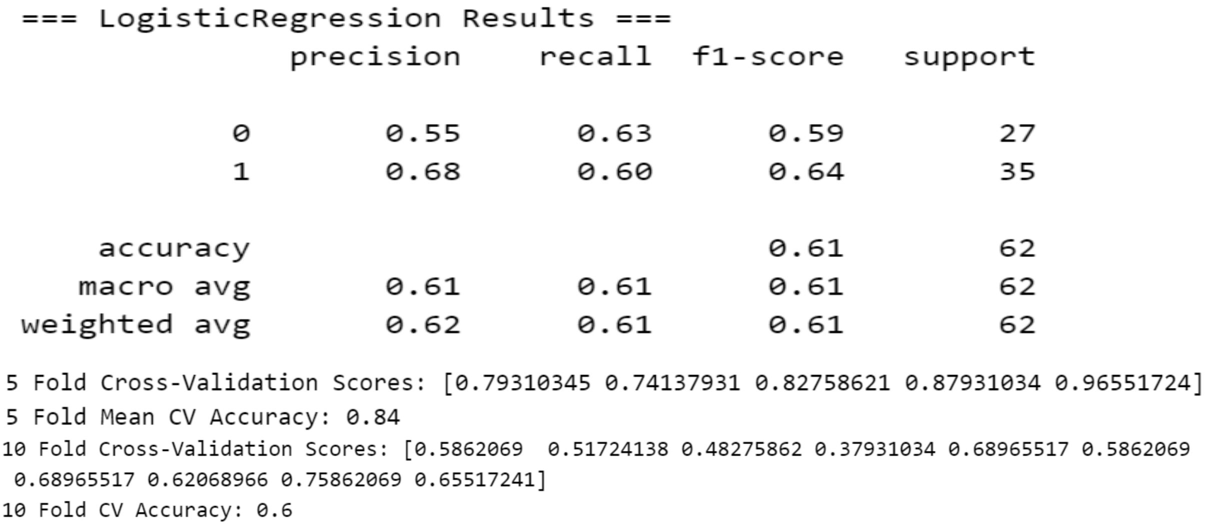
- Test Accuracy: 61%
- Cross-Validation Accuracy (5-fold): 60%
- Cross-Validation Accuracy (10-fold): 60% [2][6]



```
LogisticRegression
LogisticRegression(max_iter=1000)
```

(Figure 2: Confusion Matrix of Logistic Regression)

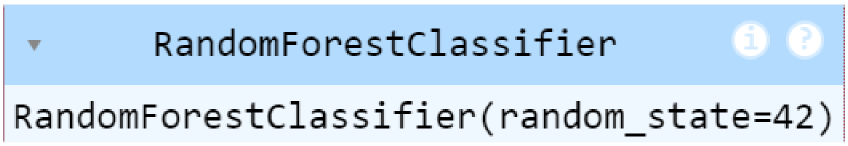
(Table 3: Classification Report & Cross validation Accuracy)



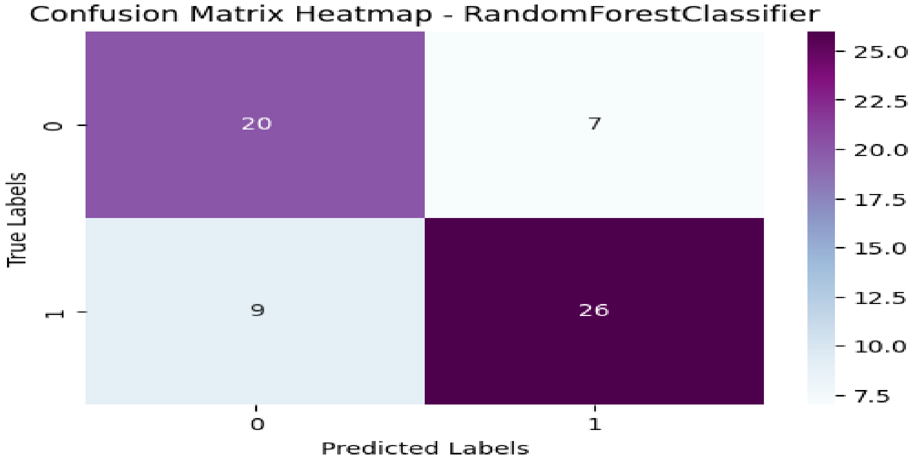
```
=== LogisticRegression Results ===
              precision    recall  f1-score   support

           0       0.55      0.63      0.59        27
           1       0.68      0.60      0.64        35

    accuracy                           0.61        62
   macro avg       0.61      0.61      0.61        62
weighted avg       0.62      0.61      0.61        62

5 Fold Cross-Validation Scores: [0.79310345 0.74137931 0.82758621 0.87931034 0.96551724]
5 Fold Mean CV Accuracy: 0.84
10 Fold Cross-Validation Scores: [0.5862069  0.51724138 0.48275862 0.37931034 0.68965517 0.5862069
 0.68965517 0.62068966 0.75862069 0.65517241]
10 Fold CV Accuracy: 0.6
```

3.2.2. Random Forest Classification

- Test Accuracy: 74%
- Cross-Validation Accuracy (5-fold): 84%
- Cross-Validation Accuracy (10-fold): 87% [3][6]

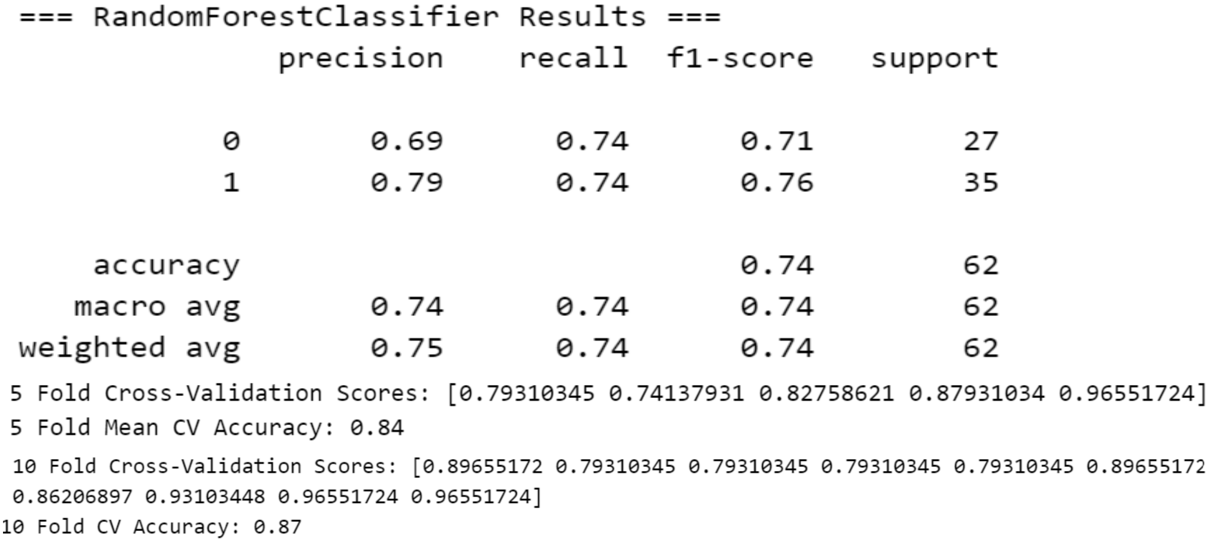


```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```



(Figure 2.1 : Confusion Matrix of Random Forest Classifier)

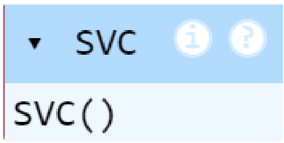
(Table 3.1: Classification Report & Cross validation Accuracy)



```
=== RandomForestClassifier Results ===
              precision    recall  f1-score   support

           0       0.69      0.74      0.71        27
           1       0.79      0.74      0.76        35

    accuracy                           0.74        62
   macro avg       0.74      0.74      0.74        62
weighted avg       0.75      0.74      0.74        62

5 Fold Cross-Validation Scores: [0.79310345 0.74137931 0.82758621 0.87931034 0.96551724]
5 Fold Mean CV Accuracy: 0.84
10 Fold Cross-Validation Scores: [0.89655172 0.79310345 0.79310345 0.79310345 0.79310345 0.89655172
 0.86206897 0.93103448 0.96551724 0.96551724]
10 Fold CV Accuracy: 0.87
```

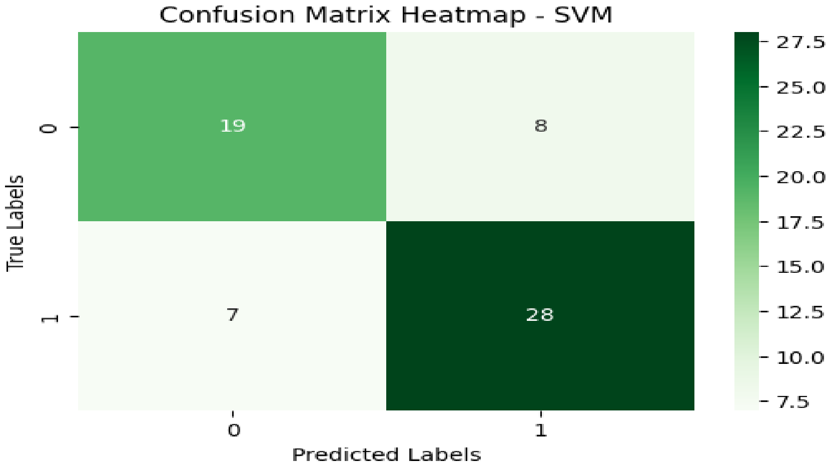
3.2.3. Support Vector Machine (SVM)

- Test Accuracy: 76%
- Cross-Validation Accuracy (5-fold): 84%
- Cross-Validation Accuracy (10-fold): 82% [4][6]



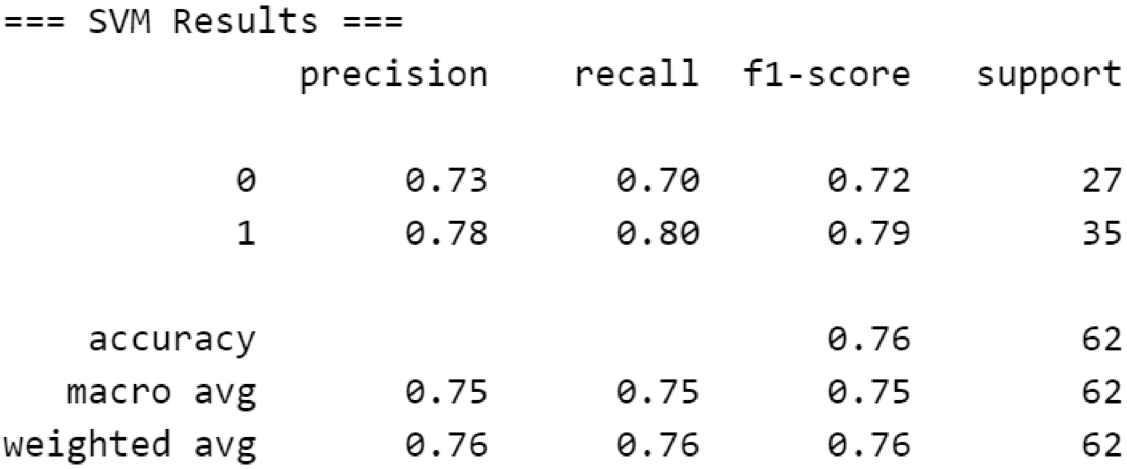
▾ SVC

```
SVC()
```

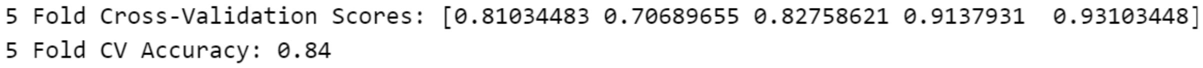


(Figure 2.2 : Confusion Matrix of SVM)

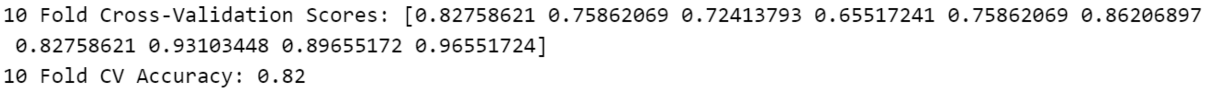
(Table 3.2 : Classification Report & Cross validation Accuracy)



```
=== SVM Results ===
              precision    recall  f1-score   support

           0       0.73      0.70      0.72        27
           1       0.78      0.80      0.79        35

    accuracy                           0.76        62
   macro avg       0.75      0.75      0.75        62
weighted avg       0.76      0.76      0.76        62
```



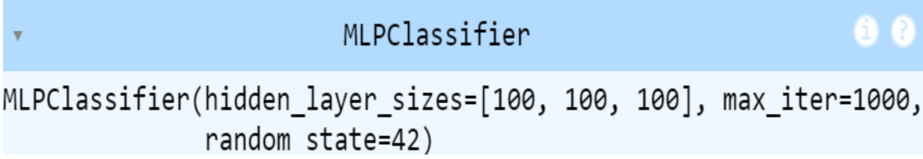
```
5 Fold Cross-Validation Scores: [0.81034483 0.70689655 0.82758621 0.9137931  0.93103448]
5 Fold CV Accuracy: 0.84
```



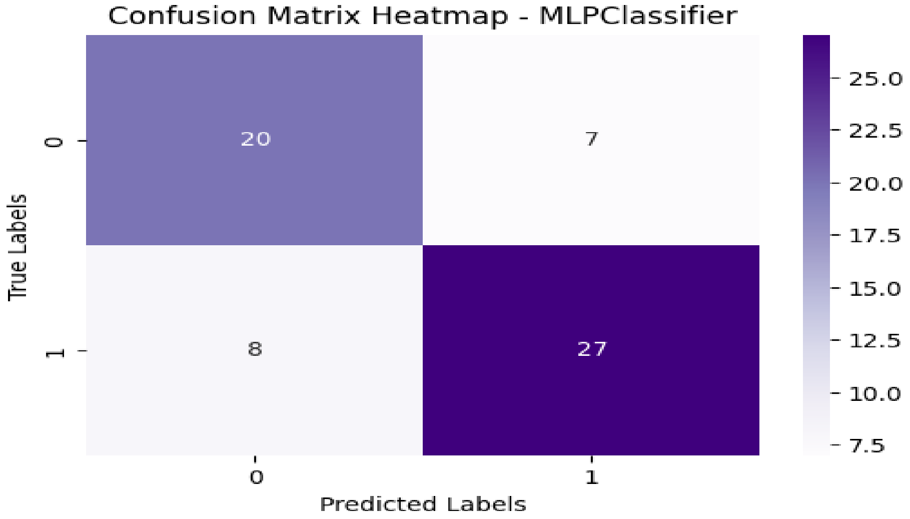
```
10 Fold Cross-Validation Scores: [0.82758621 0.75862069 0.72413793 0.65517241 0.75862069 0.86206897
 0.82758621 0.93103448 0.89655172 0.96551724]
10 Fold CV Accuracy: 0.82
```

3.2.4. Multilayer Perceptron (MLP)

- Test Accuracy: 76%
- Cross-Validation Accuracy (5-fold): 86%
- Cross-Validation Accuracy (10-fold): 86% [5][6]

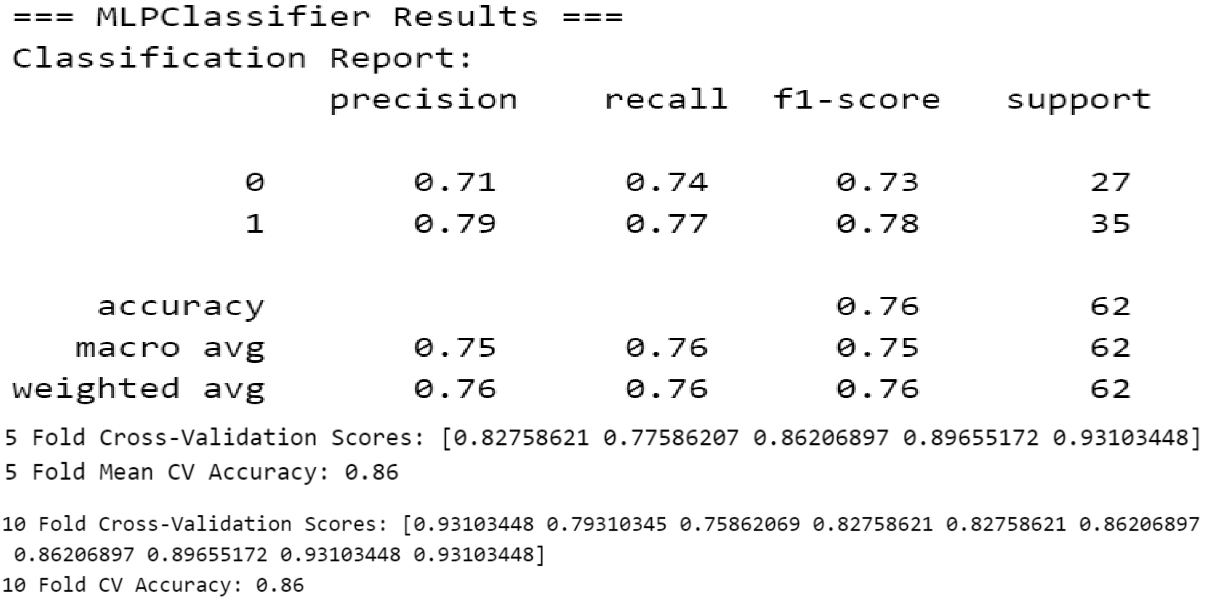


```
MLPClassifier
MLPClassifier(hidden_layer_sizes=[100, 100, 100], max_iter=1000,
              random_state=42)
```



(Figure 2.3 : Confusion Matrix of Random Forest Classifier)

(Table 3.2 : Classification Report & Cross validation Accuracy)



```
=== MLPClassifier Results ===
Classification Report:
              precision    recall  f1-score   support

           0       0.71      0.74      0.73        27
           1       0.79      0.77      0.78        35

    accuracy                           0.76        62
   macro avg       0.75      0.76      0.75        62
weighted avg       0.76      0.76      0.76        62
5 Fold Cross-Validation Scores: [0.82758621 0.77586207 0.86206897 0.89655172 0.93103448]
5 Fold Mean CV Accuracy: 0.86

10 Fold Cross-Validation Scores: [0.93103448 0.79310345 0.75862069 0.82758621 0.82758621 0.86206897
 0.86206897 0.89655172 0.93103448 0.93103448]
10 Fold CV Accuracy: 0.86
```

3.3. Conclusion

In this study, we evaluated four machine learning models Logistic Regression, Support Vector Machine (SVM), Random Forest, and Multilayer Perceptron (MLP) to predict student academic performance based on behavioural and demographic features [3][5]. The performance of each model was assessed using test accuracy and cross-validation accuracy (both 5-fold and 10-fold), which provided insights into their generalization capabilities.

The Random Forest classifier emerged as the best-performing model, achieving the highest 10-fold cross-validation accuracy of 87%, followed closely by the MLP at 86%. SVM also performed well, while Logistic Regression recorded the lowest accuracy, which is expected due to its linear nature and limited capacity to capture complex feature interactions.

The table below summarizes the performance comparison:

(Table 4: comparison of Models)

Model	**Test Accuracy**	**5-Fold CV Accuracy**	**10-Fold CV Accuracy**
Logistic Regression	61%	60%	60%
Random Forest	74%	84%	**87%**
Support Vector Machine	76%	84%	82%
Multilayer Perceptron	76%	86%	86%

These results indicate that non-linear models like Random Forest and MLP are better suited for this classification task due to their ability to model complex relationships within the data. Random Forest is recommended for future applications given its robustness, interpretability (via feature importance), and superior accuracy.

4. REFERENCES

[1] A. Romero, S. Ventura, and E. García, "Data mining in course management systems: Moodle case study and tutorial," Computers & Education, vol. 51, no. 1, pp. 368–384, 2008.

[2] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*, 3rd ed. Hoboken, NJ, USA: Wiley, 2013.

[3] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[4] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.

[5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, Cambridge, MA, USA: MIT Press, 2016.

[6] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. 14th Int. Joint Conf. Artif. Intell.*, Montreal, Canada, 1995, pp. 1137–1143.

[7] T. Fawcett, "An introduction to ROC analysis," Pattern Recognition Letters, vol. 27, no. 8, pp. 861–874, 2006.

[8] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, Cambridge, U.K.: Cambridge Univ. Press, 2014.

[9] Scikit-learn: LabelEncoder Documentation, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

[10] J. G. Skellam, "The frequency distribution of the difference between two Poisson variates belonging to different populations," J. R. Stat. Soc. Ser. A, vol. 109, pp. 296, 1946.

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.

[12] Scikit-learn: StandardScaler Documentation, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html