Algorithm1: Calculating Target Variables

1. Assign values to variables

a = 1-1 semester percentage, b= 1-2 semester percentage
c= 2-1 semester percentage, d = 2-2 semester percentage
e= 3-1 semester percentage, f = 3-2 semester Percentage
g = Attendance percentage, h = extracurricular activities
i = Academic awards and achievements, j = Coding skills
k = semester_grades=[a,b,c,d,e,f]

2.Calculate dropout
dropout = 1 if min(k) < 35 and g < 30 else 0

3.Calculate good performance
good_performance = 1 if all(grade > 60 for grade in k) else 0

4.Calculate poor performance
poor_performance = 1 if max(k) < 40 else 0

5.Calculate support required
support_required = 1 if any(40 <= grade < 60 for grade in k) else 0

6.Calculate eligibility for placement
eligible_for_placement = 1 if all(grade > 65 for grade in k) and (j or i or h) else 0

Algorithm 2: LSTM for Student Academic Performance Evaluation:

Description: This algorithm analyzes student performance using an LSTM model. The input data includes student academic performance metrics, and the output is a prediction of good performers, poor performers, students who require support and the dropouts

Input: semester percentages

Output: A binary classification indicating whether a student is a good performer or bad performer or requiring support or dropout

Procedure:

1. Import required libraries for data analysis, data cleaning, visualization, and LSTM modeling.
2. Load the dataset of student academic performance metrics
3. Clean the data
4. Evaluate the academic performance of good performers, poor performers, student who require support, student dropouts, students with placement eligibility by calculating their cumulative percentage and display the output.
5. Visualize the critical values as graphs across all students
6. Visualize the critical values as graphs across all students
7. Prepare the data for the LSTM model by separating input features and the target variable.
8. Split the data into training and testing sets
9. Reshape the input data into the required format for LSTM modeling.
10. Build and compile the LSTM model
11. Train the LSTM model on the training data.
12. Evaluate the LSTM model and print accuracies of trained LSTM model.