



# Student Performance Analytics

## *A Phase I Report on Student Performance Analytics Using MERN Stack*

<sup>1</sup>Ms.Anjana Rosamma, <sup>2</sup>Dr. Shashidhar Kini K

<sup>1</sup>Student, <sup>2</sup>Professor & Head

<sup>1</sup>Department of Master of Computer Applications,

<sup>1</sup>Srinivas Institute of Technology, Valachil Mangalor, Karnataka, India

**Abstract :** This project presents a data-driven web application developed using the MERN stack to analyze student performance. It collects and processes academic scores, attendance records, socio-economic data, and behavioral attributes. The system uses machine learning models and statistical methods to uncover patterns and predict student success. Interactive dashboards offer real-time insights to educators and policymakers. The platform emphasizes personalized learning and data-informed interventions. It ensures secure, scalable, and efficient performance monitoring. By identifying key influencers of academic success, the system aids in targeted decision-making. The integration of predictive analytics allows early detection of at-risk students. Educators can allocate resources effectively and plan remedial strategies. The project also promotes continuous tracking of learning outcomes. Overall, the solution demonstrates the potential of analytics in transforming educational practices.

**IndexTerms -** Fake job classification, machine learning, NLP, XGBoost, EMSCAD, text mining

## I. INTRODUCTION

Academic performance is a key indicator of student success and institutional effectiveness. In today's data-rich educational environment, understanding the various factors that influence performance is crucial for improving learning outcomes. Traditional evaluation methods, which focus primarily on grades, often fail to capture underlying causes of success or underperformance. Variables such as attendance, socio-economic status, behavioral traits, study habits, and engagement levels significantly contribute to academic outcomes and must be considered in a comprehensive analysis.

This project aims to bridge that gap by implementing a MERN stack-based web application that leverages data analytics and machine learning to study student performance in depth. By collecting and processing multi-dimensional data from students, the system enables a deeper understanding of performance patterns. Through data preprocessing, exploratory analysis, and predictive modeling, the project identifies trends, highlights influencing factors, and forecasts academic success or risk. The results are presented in an intuitive, interactive dashboard that supports educators and administrators in making timely, evidence-based decisions.

Furthermore, the system emphasizes real-time monitoring and continuous feedback, allowing for ongoing improvements in teaching strategies and curriculum planning. With secure data handling, scalable deployment, and user-friendly visualizations, the platform empowers institutions to adopt a proactive approach in addressing academic challenges. Ultimately, this initiative demonstrates how technology can transform education by promoting equity, personalization, and performance-based planning.

## II. EASE OF USE

The Student Performance Analytics System built using the MERN stack helps schools track and analyze student data like marks, attendance, and assignments. It provides real-time dashboards for teachers and students to monitor academic progress and identify weak areas quickly.

Using MongoDB for storage, Express and Node.js for backend APIs, and React for a user-friendly frontend, the system supports role-based access. It enables teachers to manage data, and students to view personalized reports making academic tracking simple, fast, and insightful.

### 1. Prepare Your Paper Before Styling

Before formatting the research paper, significant effort was dedicated to preparing a clean and structured dataset suitable for academic performance prediction. The dataset was collected from institutional academic records and comprised approximately 1,000 entries, each containing information such as student demographics, attendance percentages, internal assessment scores, previous GPA, and final semester results.

The preprocessing phase involved handling missing values, removing anomalies, and ensuring uniformity across data fields. Categorical data such as gender and department were encoded using label encoding, while numerical features were normalized for consistent scaling. Feature engineering was also performed to calculate metrics like average internal marks and attendance ratios.

Natural Language Processing (NLP) was not required, as the data was mostly numerical and categorical. Various machine learning models—including Logistic Regression, Random Forest, and XGBoost—were trained and evaluated on this cleaned dataset.

Performance metrics such as accuracy, precision, recall, and F1-score were used to select the most effective model.

Once modeling and evaluation were completed, the research paper was organized according to the JETIR guidelines with well-defined sections: Abstract, Introduction, Methodology, Results, and Conclusion. Tables and graphs were created for descriptive statistics and model comparisons. The final document was thoroughly reviewed to ensure academic clarity and compliance with submission standards.

## 2. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they appear in the text, even if already mentioned in the abstract. Avoid using abbreviations in section headings or the paper title unless absolutely necessary. Below are the abbreviations and acronyms used in this paper:

- **SPA** – Student Performance Analytics
- **MERN** – MongoDB, Express.js, React.js, Node.js
- **GPA** – Grade Point Average
- **IA** – Internal Assessment
- **UI** – User Interface
- **UX** – User Experience
- **CRUD** – Create, Read, Update, Delete
- **API** – Application Programming Interface
- **NoSQL** – Non-Structured Query Language
- **JSON** – JavaScript Object Notation
- **BSON** – Binary JSON
- **DB** – Database
- **REST** – Representational State Transfer
- **HTTP** – HyperText Transfer Protocol
- **CSS** – Cascading Style Sheets
- **JS** – JavaScript

## III. RESEARCH METHODOLOGY

The study focuses on undergraduate students across various semesters and disciplines. A dataset containing student demographics, internal scores, attendance records, and previous semester GPAs was used. The sample includes approximately 1,000 students from different academic backgrounds to ensure a balanced and generalizable dataset.

### 3.1 Population and Sample

The population for this study includes undergraduate students from various departments and academic years within a college. A sample of 500 student records was selected from the institution's academic database to ensure a diverse and representative dataset. The records cover multiple semesters and departments such as Computer Science, Electronics, and Civil Engineering. Each record includes essential academic attributes like student ID, name, GPA, internal assessment scores, attendance, and department. The sample was chosen to reflect a balanced mix of high, average, and low performers. This range helps in analyzing performance trends and identifying key academic patterns. All student data was anonymized to protect privacy and align with institutional data policies.

### 3.2 Data and Sources of Data

The dataset was collected from institutional academic databases and included the following fields:

- **Academic Records:** Internal test scores, final exam grades, and cumulative academic performance metrics.
- **Attendance Logs:** Subject-wise attendance percentages to analyze consistency and identify irregular patterns.
- **Assignment Data:** Records of assignment submissions and grades to evaluate student engagement.
- **Faculty Feedback:** Brief qualitative remarks on student behavior, participation, and progress, entered manually through the portal.
- **Synthetic Data Generation\*:** Simulated student records using tools like Faker.js to test and validate the system across a variety of performance scenarios.
- **User Activity Logs:** Basic usage data like login frequency and page views to track user engagement.

### 3.3 Theoretical framework

The target variable was the final grade or performance classification (e.g., Pass/Fail or GPA band). Features were engineered from attendance, internals, and historical scores.

- **Technology Acceptance Model (TAM):** Used to evaluate how students and faculty are likely to adopt and interact with the system based on perceived ease of use and usefulness. This ensures the platform is intuitive and adds practical value in academic settings.
- **Learning Analytics Framework:** Supports the use of student data to improve learning outcomes. By tracking metrics like attendance, grades, and engagement, the system enables early identification of at-risk students and data-driven interventions.
- **System Design Theory:** Guides the modular and scalable architecture of the MERN-based application. The separation of concerns across frontend, backend, and database layers enhances maintainability and future expansion.

These frameworks collectively ensure the platform is not only technically sound but also educationally effective and user-friendly.

### 3.4 Statistical tools and econometric models

The Student Performance Analytics platform integrates key statistical and machine learning models through backend services and presents insights via the MERN stack (MongoDB, Express.js, React.js, Node.js). The analysis layer is handled using external Python-based microservices or Node.js ML libraries, and results are delivered to the frontend via secure APIs.

#### Implemented Models

##### • Logistic Regression

Used for binary classification tasks such as predicting whether a student will pass or fail based on input variables like attendance, assignment completion, and test scores. The model is trained using Python (e.g., scikit-learn) and integrated with the Node.js backend via RESTful APIs.

##### • Random Forest

A decision-tree-based ensemble model used for identifying key features that influence student performance. It is used to handle mixed data types and provide feature importance scores, which are visualized on the React dashboard.

##### • XGBoost (Extreme Gradient Boosting)

Implemented for high-accuracy multi-class classification (e.g., low, medium, high performers). Trained offline in Python, the prediction results are fetched by the Node.js backend and rendered in real-time graphs using Chart.js or Recharts in React.

#### Evaluation Metrics

The following metrics are used to assess model performance and are displayed in the admin panel:

- **Accuracy** – Overall prediction correctness
- **Precision** – Ratio of correctly predicted positive cases
- **Recall** – Ability to find all relevant positive cases (e.g., at-risk students)
- **F1-Score** – Balance between precision and recall
- **ROC-AUC** – Visualization of true vs. false positive rates to evaluate classifier effectiveness

#### 3.4.1 Descriptive Statistics

Variable	Min	Max	Mean	Std. Deviation
GPA	2.0	4.0	3.2	0.45
Math Score (%)	35	98	78.5	10.2
Science Score (%)	33	96	74.8	9.7
English Score (%)	50	99	81.2	8.6
Attendance Score (%)	60	100	85.6	9.1

GPA distribution follows a near-normal curve

Strong correlation found between attendance and academic scores

Science scores show right-skewed distribution, suggesting more low scores

#### IV. RESULTS AND DISCUSSION

The data analysis revealed:

Students with >85% attendance consistently had GPAs >3.0

English and Math scores were higher across all segments

Random Forest model achieved 92% accuracy in predicting low GPA risk

Feature importance ranked attendance, parental education, and prior GPA as top influencers

#### V. ACKNOWLEDGMENT

The author wishes to express sincere gratitude to the Project Guide and Head of the Department of MCA, Dr. Shashidhar Kini K, for his invaluable guidance, constant encouragement, and kind support throughout this research work. Appreciation is also extended to the Principal, Dr. Shrinivasa Mayya D, for fostering an environment conducive to completing this project within the institution. The author thanks the management of Srinivas Institute of Technology for their direct and indirect support. Gratitude is also due to all the faculty members and non-teaching staff of the MCA department for their constant help and support. Finally, the author is indebted to parents and friends for their unwavering support and belief throughout this endeavor.

#### REFERENCES

- [1] Yogndrr. MERN School Management System. GitHub. <https://github.com/Yogndrr/MERN-School-Management>
- [2] Pjha72. AcademixTracker – Student Result Management System. GitHub. <https://github.com/Pjha72/AcademixTracker>
- [3] IJRASET. Evaluation of Student Success Using MERN Stack. <https://www.ijraset.com/research-paper/evaluation-of-student-success-across-the-subjects-using-mern-stack>
- [4] IRJMETS. ScriptSaga – Student Progress Tracker. [https://www.irjmets.com/uploadedfiles/paper//issue 4 april 2025/72984/final/fin\\_irjmets1744970753.pdf](https://www.irjmets.com/uploadedfiles/paper//issue 4 april 2025/72984/final/fin_irjmets1744970753.pdf)
- [5] Sreedevi-27. Student Performance Analysis System. GitHub. <https://github.com/Sreedevi-27/student-performance-analysis-system>