International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

AI-Driven Approach to Student Performance Analysis System

Rajat Srivastava[1], Mr. Ankit Singh[2], Sneha Mehrotra[3], Shaifali Singh[4], Shreyansh Srivastav[5]
[1,3,4,5]Computer Science and Engineering, (Data Science) Babu Banarasi Das Institute of Technology and Management (Dr. A.P.J. Abdul Kamal Technical University) Lucknow, India.
[2]Information Technology Babu Banarasi Das Institute of Technology and Management (Dr. A.P.J. Abdul Kamal Technical University) Lucknow, India

Abstract- There's a lot more to student performance than just marks. Some kids barely pass written exams but shine in group projects or sports. The problem is, most colleges still judge students almost entirely by their test scores. That's like judging a fish by its ability to climb a tree. By the time a teacher realizes someone's struggling, that student might already be failing or even thinking of dropping out. So what if we could spot trouble earlier — way before the report card says it all? That's what this paper is about. We used machine learning to sift through student data — attendance, past grades, even family background — and predict who might fall behind. Not just for the sake of prediction, but to actually give teachers a heads-up so they can step in and help. The results were pretty solid. Our model caught most at-risk students with over 90% accuracy. Not perfect, but a lot better than waiting till the end of the semester.

Keywords- Student performance, Machine learning, Data mining, Academic prediction, Early warning system, Learning analytics.

I. INTRODUCTION

Let's be honest — education today is still very exam-obsessed. You write a test, you get a number, and that number decides if you're —good student‖ or not. But anyone who's spent time in a classroom knows it's never that simple. Some students are terrible at memorizing formulas but great at solving real-world problems. Others freeze during exams but participate actively in class discussions. None of that shows up in a grade sheet.

Meanwhile, colleges are sitting on mountains of data — attendance records, assignment submissions, library logins, even canteen usage patterns in some places. Yet most of it just sits in some database, collecting dust. We don't use it to actually help students until it's too late.

This project started from a simple question: instead of waiting for students to fail, why not use all that data to predict who's heading that way? Machine learning felt like the obvious tool. It's already being used to recommend movies, detect credit card fraud, even diagnose diseases. So why not apply it to something as fundamental as education?

We're not saying a model can replace a teacher's instinct. But it can highlight patterns humans might miss — like a sudden drop in assignment submission frequency, or a steady decline in attendance that started weeks ago. Those are red flags. Our system tries to flag them early, so teachers can actually do something about it.

II. LITERATURE SURVEY

A. The Old Way of Grading

For decades, student evaluation meant looking at exam scores and maybe class participation. It was backward-looking. You'd find out someone failed only after they'd already failed. No one thought about predicting failure — or even considered that home environment or sleep patterns might matter.

B. Data Mining in Education

Around the late 2000s, researchers started experimenting with algorithms on student data. Decision trees, neural nets — they all showed promise. Some studies managed to predict dropouts with decent accuracy. But most stayed inside research papers. Hardly anyone implemented them in actual colleges.

C. Learning Analytics Platforms

Recently, a few universities have rolled out dashboards that track student engagement. You log in, see a heatmap of who's attending class, who's submitting assignments late. It's a step forward, but still mostly descriptive. It tells you what is happening, not what's about to happen.

D. Where They Fall Short

Most existing systems are either too complex to scale, or too opaque — you get a prediction but no explanation. Also, very few of them trigger automatic alerts. A teacher still has to manually check the dashboard. That defeats the purpose of early intervention. We wanted to build something that doesn't just predict, but actively notifies.

III. METHODOLOGY

Our pipeline was straightforward: collect data, clean it, train a model, test it, then see if the predictions made sense.

A. How the System is Built

We kept the architecture modular. You can swap out the ML engine later if something better comes along. The system has five layers:

1. Data input — pulls records from college databases.
2. Preprocessing — fixes missing values, encodes text fields.
3. ML engine — trains on historical data.
4. Prediction module — classifies students into performance bands.
5. Reporting interface — shows results to teachers.

Nothing fancy, just clean separation of concerns.

Where We Got the Data

We used a mix of internal student records and public datasets from UCI and Kaggle. The features we tracked:

- Attendance percentage
- Internal assessment scores
- Previous semester SGPA
- Self-reported study hours per week
- Parents' highest education level
- Participation in extracurricular activities (yes/no)

We made sure to anonymize everything — no names, no roll numbers. Just rows of numbers and categories.

Table I: What We Tracked

Category	Examples
Academic	Internal marks, past grades
Behavioral	Attendance, study hours
Demographic	Age, gender, parental education
Engagement	Assignments done, events joined

Cleaning the Mess

Real-world data is never clean. Some students hadn't updated their profiles — missing parent education fields. Others had attendance over 100% (clearly a data entry error). We fixed missing values with averages or the most common value. Text labels like ―Male‖ /―Female‖ became 0 and 1. Then we scaled everything so no single feature dominated just because its numbers were bigger.

We also removed outliers — like that one student with 150% attendance. Then we split the cleaned dataset: 80% for training, 20% kept aside for final testing.

Choosing the Right Model

We tested four algorithms: Decision Tree, Random Forest, SVM, and Logistic Regression. Random Forest outperformed the rest — better accuracy, less overfitting. Probably because it averages multiple trees, so it's not thrown off by weird data points.

We trained it on past student records labeled as Excellent, Good, Average, or Poor. Those labels were based on final SGPA brackets.

Making Predictions

Once trained, the model takes a new student's data and outputs a performance category. We specifically look for ―Poor‖ and sometimes
―Average‖ flags. Those are the ones who need attention. The system then sends a simple alert to the faculty dashboard. No complex charts — just a list of names and suggested action:
―Schedule a meeting‖, ―Extra tutoring‖, ―Check attendance‖.

IV. RESULTS AND EVALUATION

A. How We Tested

We coded everything in Python — pandas for data, scikit-learn for models. The test set was never touched during training. No data leakage.

B. Metrics We Used

We checked four numbers:

- **Accuracy —** overall correctness.
- **Precision** — when we say a student is at risk, how often are we right?
- **Recall —** of all the actual at-risk students, how many did we catch?
- **F1-score** — balance between precision and recall.

These matter because if we miss too many at-risk students, the system is useless. If we flag too many false positives, teachers will ignore it.

What We Found

Random Forest gave 91.8% accuracy. More importantly, recall for the ―Poor‖ category was high — over 89%. That means we caught nearly

9 out of 10 students who actually ended up struggling. A few borderline cases were misclassified — students whose performance was inconsistent. But overall, the model was reliable enough for real-world use.

We compared it with the other algorithms. SVM came second at 87% accuracy. Decision Tree was around 82%. Logistic Regression lagged at 79%. So Random Forest was the clear winner.

V. WHAT'S NEXT

This is just version one. There's room to grow:

- **Deep learning —** maybe a small neural net could squeeze out a bit more accuracy.
- **Live LMS data** — if we pull data in real time from Moodle or Blackboard, predictions could update every week.
- **Recommendation engine** — instead of just flagging students, suggest specific resources (videos, practice problems) based on their weak areas.
- **Mental health indicators —** tricky to collect, but things like sleep patterns, stress levels, social interaction frequency could add another dimension.
- **Mobile app —** most teachers check their phones more than their desktops. A simple app with push alerts would be far more useful than a web dashboard.

VI. CONCLUSION

So here's what we ended up with: a system that looks at a bunch of student data — some academic, some behavioral, some demographic — and predicts, with decent accuracy, who's likely to struggle. It's not magic. It's just pattern recognition. But it works.

The bigger point isn't the model itself. It's that colleges now have a way to stop guessing and start acting. Instead of waiting for failure to happen, they can spot the warning signs weeks or even months earlier. That changes everything — for the student, for the teacher, for the institution.

We're not claiming this solves every problem. Some students will always slip through. And a model is only as good as the data it's trained on. But compared to the old system — where no one noticed a student was struggling until they failed the final exam — this is a massive improvement.

Education has always been about helping people grow. It's time our tools caught up with that mission.

Acknowledgment

We owe a lot to our professors at BBDITM, especially those who let us pick their brains during the early stages of this project. Their feedback saved us from some pretty dumb mistakes.

Also, huge thanks to the open-source community. Kaggle and UCI gave us free access to datasets we couldn't have collected ourselves. Scikit-learn and pandas did all the heavy lifting. This project stands on the shoulders of giants — or at least, on the shoulders of people who write really good documentation.

REFERENCES

1. R. Baker and K. Yacef, ―The State of Educational Data Mining in 2009,‖ Journal of Educational Data Mining, vol. 1, no. 1, pp. 3–17, 2009.
2. T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. Springer, 2017.
3. C. Romero and S. Ventura, ―Educational Data Mining: A Review of the State of the Art,‖ IEEE Transactions on Systems, Man, and Cybernetics, Part C, vol. 40, no. 6, pp. 601–618, Nov. 2010.

4. S. B. Kotsiantis, —Use of Machine Learning Techniques for Educational Purposes: A Decision Support System for Forecasting Students' Grades,‖ Artificial Intelligence Review, vol. 37, no. 4, pp. 331–344, 2012.
5. A. Peña-Ayala, —Learning Analytics: A Review of the Literature,‖ in Learning Analytics: Fundaments, Applications, and Trends, A. Peña-Ayala, Ed. Springer, 2018, pp. 1–22.
6. G. Siemens and R. S. J. d. Baker, —Learning Analytics and Educational Data Mining: Towards Communicatio and Collaboration,‖ in Proc. 2nd Int. Conf. Learning Analytics and Knowledge, 2012, pp. 252–254.
7. A. Wolff, Z. Zdráhal, A. Nikolov, and M. Pantucek, —Improving Student Retention: A Machine Learning Approach to Predicting At-Risk Students,‖ in Proc. IEEE Int. Conf. Advanced Learning Technologies, 2013, pp. 145–149.
8. J. R. Mathew, A. Kuriakose, and R. S. Hegde, —Student Performance Prediction Using Machine Learning Techniques,‖ in Proc. IEEE Int. Conf. Advances in Computing, Communications and Informatics (ICACCI), 2017, pp. 264–269.
9. S. Hussain, F. Zhu, Z. Zhang, and R. Abidi, —Predicting Student Engagement in E-Learning and Its Impact on Academic Performance,‖ Computational Intelligence and Neuroscience, vol. 2018, Article ID 8458972, 2018.
10. Y. Abu Amrieh, T. Hamtini, and I. Aljarah, —Preprocessing and Analyzing Educational Data Set Using X-APIs for Improving Student's Performance,‖ International Journal of Computer Science and Information Security, vol. 13, no. 8, pp. 35– 40, 2015.