www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 09, September 2025, pp : 56-58

e-ISSN : 2583-1062
Impact Factor : 7.001

PREDICTING THE ACADEMIC PERFORMANCE OF UNDERGRADUATE COMPUTER SCIENCE STUDENTS USING DATA MINING

Shubha R[1]

[1]Student Of MCA, School Of Applied Science, Sapthagiri NPS University Bengaluru, India.

E-Mail: shubhaaradya987@gmail.com

ABSTRACT

Predicting how well a student might do academically is something educators and institutions are always working on. There are a bunch of factors—personal, academic, even environmental—that can influence student performance, and figuring out how they all come together isn't easy. In this project, I used data mining techniques to try and make sense of it. The idea was to analyze past data to see if we can predict students' final academic outcomes. I worked with a real-world dataset of undergraduate students, and applied different classification algorithms—Decision Trees, Naïve Bayes, and k-Nearest Neighbors (KNN)—to see which one worked best. I ran the models using 10-fold cross-validation to make sure the results were reliable. Among all the algorithms, the Decision Tree turned out to give the highest accuracy. So, based on this, I believe data mining can be a really helpful tool in forecasting academic performance, and it could eventually help educators take early action for students who might be struggling.

1. INTRODUCTION

Nowadays, there's a growing focus on making sure students not only enroll in higher education but also actually succeed in it. With education becoming more competitive and complex, it's important to figure out early on which students might be at risk of underperforming. This way, schools and colleges can step in and offer support before things get worse.

A student's academic performance depends on many things—personal background, academic history, family situation, social environment, even psychological factors.

Because all of these can interact in complicated ways, predicting academic success isn't straightforward. That's where data mining comes in. It gives us tools to analyze large sets of student data and find useful patterns that we might miss otherwise.

In this project, I used data mining to try and predict how students would perform at the end of their academic program. My goal was to build a predictive model that could help identify students who are more likely to perform poorly, so that timely interventions could be made.

I worked with a dataset collected from undergraduate students. After cleaning and preparing the data, I applied three machine learning classification algorithms: Decision Tree (J48), Naïve Bayes, and k-Nearest Neighbors (KNN). These models were trained and tested using 10-fold cross-validation, which helps make sure the results are not biased or overly specific to a certain part of the data.

In the end, the Decision Tree algorithm gave the best results in terms of accuracy, precision, recall, and F-measure. The findings show that it's definitely possible to predict student performance using machine learning—and that these kinds of models can play a real role in improving educational outcomes if used correctly.

2. RELATED WORK

There's been quite a lot of research into predicting how students perform using different data mining techniques. Researchers have experimented with many algorithms like Decision Trees, Naïve Bayes, Support Vector Machines (SVM), Artificial Neural Networks, and even more complex ensemble models.

For example, Pandey and Pal (2011) used decision trees to predict student performance in different subjects. They found that decision trees worked well for spotting students who might be at risk. Another study by Al-Barrak and Al-Razgan (2016) also compared different classification models, and again decision trees came out on top in terms of accuracy.

Kotsiantis et al. (2004) applied Naïve Bayes, Decision Trees, and KNN to predict final grades and found that Naïve Bayes performed best with smaller datasets. Similarly, Hijazi and Naqvi (2006) looked at how factors like family income and parents' education levels affect student performance. They concluded that socio-economic factors play a big role.

Sembiring et al. (2011) used clustering techniques to group students based on performance levels. That helped them better understand trends and learning behavior. Also, Cortez and Silva (2008) built a predictive model using student-

www.ijprems.com
editor@ijprems.com
INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 09, September 2025, pp : 56-58
e-ISSN : 2583-1062
Impact Factor : 7.001

related data, including things like study time and alcohol use, and applied several data mining methods to find the best fit.

From all this, it's clear that data mining can be very effective in education. Most of the work done so far has focused on using machine learning techniques to predict grades or identify students who might drop out. The methods vary, but many studies show that classification algorithms like Decision Trees, Naïve Bayes, and KNN consistently give good results.

In my project, I've built on these past studies, using a real dataset from undergrad students and focusing on classification algorithms that have already proven to be effective in similar work.

3. METHODOLOGY

For this project, I followed a step-by-step approach that involved collecting data, preparing it, choosing the right algorithms, training the models, and evaluating their performance. Below is an overview of how I carried everything out.

3.1 Data Collection

The dataset I used contains academic records of undergraduate students from a university. It includes different types of information like personal details, academic background, and final grades. Each record represents one student and includes both input features and a final result (pass/fail) as the target variable.

3.2 Data Preprocessing

Before feeding the data into any algorithm, I cleaned and prepared it to make sure the models would work well. This step included:

- Handling missing or incomplete data
- Converting categorical data into numerical format where needed
- Normalizing the data to keep everything on a similar scale
- Removing any duplicate or irrelevant features

I also made sure the class labels (i.e., student outcomes) were balanced, or at least not heavily skewed, so that the models wouldn't be biased.

3.3 Feature Selection

Not all the features in the dataset are equally important. So, I used feature selection techniques to identify the most useful ones. This helped reduce the noise in the data and improved both the accuracy and speed of the models.

3.4 Classification Algorithms

I experimented with three commonly used classification algorithms:

- Decision Tree (J48): A popular algorithm that creates a tree-like structure based on the features. It's easy to interpret and usually gives solid results.
- Naïve Bayes: Based on probability theory, this one assumes that all features are independent. Even though that's rarely true in real life, it still tends to perform well in many cases.
- k-Nearest Neighbors (KNN): This one classifies a new data point by looking at its 'k' closest neighbors in the dataset. It's a simple, intuitive algorithm, but can be slow with large datasets.

All models were trained and tested using 10-fold cross-validation to make sure the results were consistent and not just luck. This means the dataset was split into 10 parts—each time, 9 parts were used for training and 1 part for testing, and this process was repeated 10 times.

3.5 Evaluation Metrics

To compare the models, I used the following metrics:

- Accuracy: How often the model correctly predicted the outcome
- Precision: How many of the predicted "positive" results were actually correct
- Recall: How many actual "positive" cases the model successfully found
- F-Measure: The harmonic mean of precision and recall (a balanced score)

These metrics gave me a well-rounded view of how each algorithm performed.

4. RESULTS AND DISCUSSION

After training and testing the three classification algorithms (Decision Tree, Naïve Bayes, and KNN) on the student dataset, I compared their performance using the evaluation metrics I mentioned earlier—accuracy, precision, recall, and F-measure.

The models were all tested using 10-fold cross-validation, which helped give a fair and balanced view of how each one performed across different parts of the dataset.

Here's a breakdown of the results:

- Decision Tree (J48):
  - Accuracy: 91.90%
  - Precision: 0.919
  - Recall: 0.919
  - F-Measure: 0.919
- Naïve Bayes:
  - Accuracy: 88.99%
  - Precision: 0.890
  - Recall: 0.890
  - F-Measure: 0.890
- K-Nearest Neighbors (KNN):
  - Accuracy: 86.93%
- Precision: 0.869
  - Recall: 0.869
  - F-Measure: 0.869

As you can see, the Decision Tree model gave the best overall results across all metrics. It was not only the most accurate, but it also had the best balance between precision and recall. This means it did a good job at correctly identifying both students who passed and those who failed.

The Naïve Bayes algorithm also performed quite well, especially considering its simplicity. It's known for working well on smaller datasets or when the data isn't too noisy, which matches what I observed here.

KNN had the lowest scores among the three, possibly because it relies on distance metrics and can be affected by irrelevant or redundant features—something I did try to minimize during preprocessing, but it still may have influenced the results. These results confirm that the choice of algorithm really matters, and in this case, the Decision Tree was clearly the best option for this kind of dataset. Its ability to model non-linear relationships and provide human-readable decision paths makes it especially useful in an educational setting, where you might want to explain why a certain prediction was made. In short, this experiment shows that with the right preparation and model choice, data mining can be a powerful way to predict student performance and potentially help institutions take early action for students who are at risk.

5. CONCLUSION

In this project, I set out to see if machine learning could help predict how students would perform academically. After working with real student data and applying three different classification algorithms—Decision Tree, Naïve Bayes, and KNN—I found that the Decision Tree gave the best results in terms of accuracy, precision, recall, and F-measure.

This confirms that data mining can be a valuable tool in education. With the right kind of data, these models can give schools and universities early warnings about students who might be struggling. That way, educators can step in and offer support before it's too late. Of course, there's still room to improve.

For example, the dataset I used was somewhat limited—it came from one institution and had a fixed set of features. In the future, it would be helpful to include more diverse data like attendance records, learning behavior, psychological factors, and even online learning activity. This could make the models even more accurate and useful. Also, while I only tested a few basic classification algorithms, there are many other techniques like ensemble learning, deep learning, or hybrid models that might perform even better.

Exploring those could be a good next step. Overall, I believe that projects like this can make a real difference in how we understand and support student learning—especially if they're combined with real-time data and proactive academic interventions.

6. REFERENCES

[1] www.ijprems.com
[2] www.irjmets,com