International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Student Performance Prediction System Using Machine Learning Algorithm

S. Sujatha[1], B. Ganga Tejaswini[2], B. Vyshnavi Narayanamma[3], G. Sandhya[4], G. Lakshmi Venkata Pallavi[5], G. Venkata Sai Manaswini[6]

1Asst.Prof, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P
2UG Student, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P
3UG Student, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P
4UG Student, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P
5UG Student, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P
6UG Student, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P
Corresponding Author E-mail:manaswinireddy0611@gmail.com

Abstract

Schools are also focusing more on early intervention of poor performing students to enhance their success levels and decrease school dropout. The conventional evaluation tools which are mainly based on end semester tests give feedback which is also delayed and therefore restrictive in terms of initiation of timely action. In this paper, the Student Performance Prediction System is introduced, which uses the XGBoost (Extreme Gradient Boosting) classification algorithm to forecast the results of students based on a few important academic indicators in real-time. The system uses four main input parameters; attendance percentage, mid-term examination marks, internal assessment marks and laboratory performance where a feature engineering pipeline is used to scale up these features to seven model-ready features. The StandardScaler is used to make sure that there is data normalization so that all features contribute equally. The trained model divides students into three groups, PASS, AT RISK, and FAIL, and has a prediction accuracy of 96.5 as a whole. The interface is a web-based application created with the help of Streamlit that allows educators to enter the student data and get instant predictions with interactive visualizations created with Plotly, Seaborn and Matplotlib. The system also offers AI-based suggestions that establish the main area of academic deficiency faced by each student and come up with specific improvement suggestions. Results in experiments prove that attendance is the most predictive of academic risk (35% feature importance), then there is mid-term performance (28%), then laboratory scores (25%). The suggested system will make academic monitoring a proactive, data-driven early warning system instead of a reactive, post examination process and help the institutions efficiently distribute the support resources and enhance the general retention of students.

Keywords: Student Performance Prediction; XGBoost; Educational Data Mining; Machine Learning; Early Warning System; Streamlit.

1. Introduction

Student performance is one of the cornerstones of any educational system as it defines the academic growth, standards of grading and the quality of institutions. As student populations expand and their learning capabilities vary in terms of capabilities, tracking individual performance in school has been complicated. Students are diverse in terms of their learning rates, background, level of interests and their external conditions and therefore traditional means of evaluation cannot be used to monitor continuous performance (Romero & Ventura, 2010). The traditional evaluation models are more retrospective in nature. The grading is done by mid-term and final examination and the results are only assessed at the end of the semester. Although this method quantifies academic performance, it does not offer an opportunity to obtain early information about possible learning challenges. Students who start

having difficulties throughout the semester are described as being usually detected when final results are announced, which seriously limits the possibility of providing academic assistance in time (Bhardwaj and Pal, 2012). The advent of Artificial Intelligence (AI) and Machine Learning (ML) has contributed to the power of Educational Data Mining (EDM) in a significant way.

Such technologies can be used to analyze organized academic information to discover the concealed patterns of performance and foretell the results. Based on such indicators as attendance, internal test results, mid-term mark scores, and laboratory results, predictive models can categorize students into high-risk categories with a high level of accuracy. Contrary to the traditional systems, the ML-based approaches give real-time insights that help in the early detection of academic decline (Thai-Nghe et al., 2010).

The application of predictive analytics to academic systems has a few advantages: the active monitoring of the process during the semester, accurate identification of areas of weakness on the theory, practical, and engagement levels, customized intervention plans, and effective distribution of academic support services. The student performance prediction system is driven by the desire to minimize the level of underperformance, and dropout rates in schools by early detection of potential studying children. The system will turn raw academic data into actionable insights, which will help to foster an active, data-oriented learning experience and promote prompt intervention and increased student achievement rates.

2. Literature Review

2.1 Existing System

Multiple authors have examined the process of student performance prediction with the help of machine learning. Cortez and Silva (2008) used Decision Trees and Regression models to predict grades of secondary school students based on academic and demographic data, an aspect that showed that the structured educational data could well be used to predict student performance. The study by Pandey and Pal (2011) has used the Naive Bayes classification algorithm to classify students on the basis of academic divisions, and it was demonstrable that the classification methods can effectively identify high and low achiever students. A thorough review of Educational Data Mining methods was proposed by Romero and Ventura (2010) in IEEE Transactions on Systems, Man, and Cybernetics to note the increase in the significance of machine learning techniques in modeling academic performance, including Decision Trees, Neural Networks, and Support Vector Machines. Bhardwaj and Pal (2012) used Decision Tree algorithms to detect weak students at a young age and it is important to note that predictive models have practical value in enhancing academic performance. Thai-Nghe et al. (2010) applied the methods of classification and matrix factorization in order to predict student achievement and stated high accuracy of prediction. In their article, Chen and Guestrin (2016) presented XGBoost, a tree boosting system that is scalable and has since become a benchmark algorithm when performing structured classification of data.

Although the use of different machine learning models has already demonstrated good results, numerous learning institutions still use the old system of evaluation, which is mainly based on final examination marks. The majority of the available systems are merely record keeping systems and lack automated predictive analysis to identify at-risk students early. The main drawbacks of the current systems are subjective and manual assessment methodology, too slow to provide feedback and results are not available until final exams, very low scalability due to one student at a time analysis, generic insight due to final grades and identification of failure only after they have happened.

2.2 Proposed System

The Student Performance Prediction System proposed overcomes the shortcomings of the current methods by ensuring an intelligent Academic Support tool which is based on ML. It is unlike traditional systems because it uses XGBoost to analyze several academic parameters at the same time and give real-time predictive information. The system gathers student data based on a web interface built on Streamlit, verifies student data, engages in feature engineering and normalization, and sorts students into PASS, AT RISK or FAIL students based on probability scores. It also offers visualizations that can be interpreted and analyze the feature importance, which can help educators see what logic behind any prediction.

Table 1. Comparison of Existing vs. Proposed System

Feature	Traditional System	Proposed ML System
Methodology	Subjective, manual	Objective, data-driven
Speed	Months (post-exam)	Seconds (real-time)
Accuracy	Variable, error-prone	96.5% precision-driven
Scalability	Low (one at a time)	High (thousands instantly)
Proactivity	Reactive	Proactive (early warning)

3. Methodology

3.1 System Architecture

The system is based on a modular, layered architecture with four different layers separating data handling and user interaction as well as machine learning logic. The Presentation Layer, which is developed with Streamlit, handles all the interactions of the user with three views, which include a welcome, input form, and results dashboard. The Logic and Processing Layer is the interface between the UI and the ML model, which also includes a Data Processor to format the data, an Insights Engine producing recommendations and Visualization Engine to display the data as a chart implemented in Plotly and Matplotlib. The trained XGBoost model (model.pkl) and the fitted StandardScaler (scaler.pkl) are found in the Prediction Layer and they classify and normalize respectively. Data Layer All persistent storage such as training data and serialized model artifacts is managed by the Data Layer. Such a layered design provides maintainability, updatability and scalability of the system. The architecture as a whole is shown in Fig. 1.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

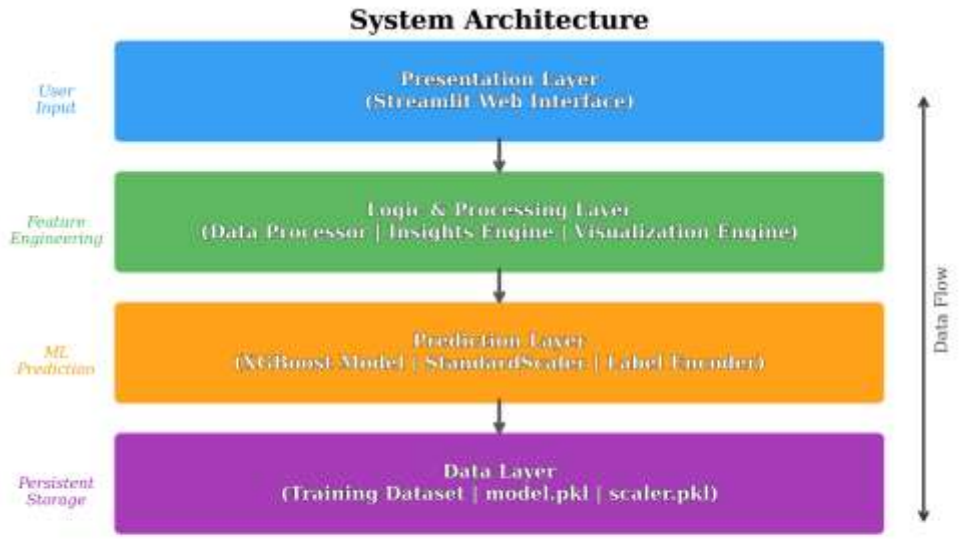

Fig. 1. Layered system architecture of the Student Performance Prediction System.

3.2 Data Processing Pipeline

The workflow of the data processing is seven consecutive steps as shown in Fig. 2. To begin with, the user input is gathered via the Streamlit interface where the educators provide the inputs in terms of attendance percentage, mid-term marks, internal assessment scores, and laboratory performance. Second, there should be data validation where all the inputs should be within acceptable limits (0-100 in terms of attendance and marks). Third, feature engineering converts the four main inputs to seven clean model features through a set of predetermined mapping logic as in Fig. 3. The attendance percent, directly proceeds to attendance, mid-term score goes in mid1 and mid2, assignment score goes in assign1 and assign2 and laboratory score goes in labinternals and labexternals with a weight of 2.33x. Fourthly, StandardScaler is a tool of data normalization that provides homogeneous feature contribution based on the following formula: $z = (x - m)/s$.

Fifth, the normalized features undergo XGBoost classifier to be predicted. Sixth, the aggregation of results deciphers the predictions into human understandable labels. Seventh, the last dashboard is generated with visualizations and AI insights.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

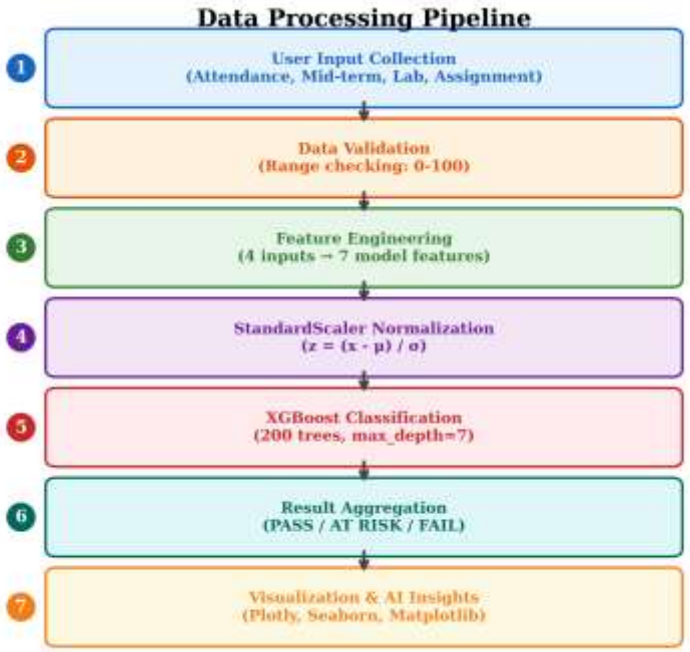

Fig. 2. Complete data processing pipeline from user input to prediction output.

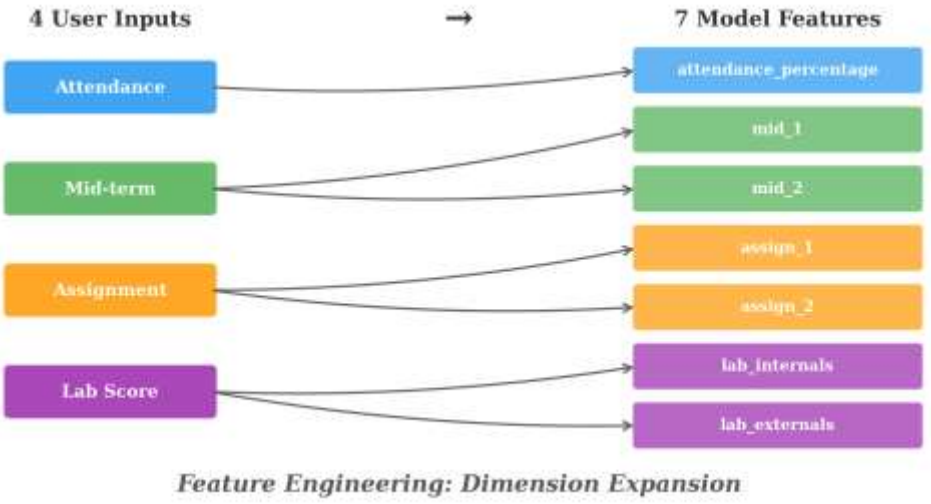

Fig. 3. Feature engineering: Mapping 4 user inputs to 7 model features.

3.3 XGBoost Classification Algorithm

XGBoost (Extreme Gradient Boosting) is a machine learning algorithm based on a decision tree which consists of gradient boosting framework. It was chosen to be used because it has a number of technical benefits: L1 and L2 regularization that help avoid overfitting to a particular student cohort, the support of sparse data, typical of student records, parallel processing, which makes it possible to conduct inferences in real time, and stable superiority in multi-class classification problems compared to more basic models, such as Logistic Regression, and standard Decision Trees (Chen and Guestrin, 2016).

The hyperparameters used to set up the model were as follows: nestimators=200 (used to build 200 trees to capture complex dependencies), maxdepth=7 (trees are not built too deep), learningrate=0.05 (conservative update step to ensure better stability) and objective= multi:softmax (optimized to estimate

multi-class outcomes). The model produces probability scores of three performance categories namely PASS, AT RISK and FAIL with the highest probability category being the final prediction.

Table 2. XGBoost Model Configuration

Parameter	Value	Purpose
n_estimators	200	Capture complex dependencies
max_depth	7	Prevent over-specialization
learning_rate	0.05	Improve training stability
objective	multi:softmax	Multi-class classification

3.4 Modules Description

Data Collection and Validation Module.

In this module, user input is processed with the help of the Streamlit interface and data integrity is ensured. It makes sure that the attendance percentage is between 0-100 percent, examination marks are within the required ranges, and all the mandatory fields are filled. Bad entries cause descriptive error messages which block the garbage in garbage out scenarios.

Feature Engineering and Mapping Module.

This module converts the four inputs by the educator to seven features that are compatible with the model. The expansion gives the XGBoost model a fine-grained perspective on the performance of students in terms of engagement, theoretical, and practical aspects. The mapping applies weighted logic in simulating realistic assessment distributions using small amounts of input data

Machine Learning Prediction Module.

The main prediction module loads the StandardScaler and XGBoost model that has been trained on the pickle file in the form of serials. It takes normalized features as input and gives out probability distributions in the three categories of classifications. Control The label decoder converts numeric predictions into status labels readable by humans.

Visualization and Reporting Module.

The module creates three categories of visual outputs, namely Plotly interactive bar charts to compare metrics side-by-side with hover effects, Seaborn statistical plots to analyze performance distribution, and Matplot library pie charts to visualize the parameter weightage. These charts offer intuitive dashboards to teachers to track the trends in cohort and individual performance.

AI Insights Module.

Post-prediction analysis module involves comparison of normalized scores of all the categories to determine the major area of academic concern to a student. It creates custom pedagogical advice based on a curated library on improvement strategies, and is able to match particular areas of weaknesses with specific guidance.

3.5 Implementation Environment

This section is about the environment within which the implementation will take place. Python (Version 3.8-3.11) is used to develop the system as it provides a wide range of machine learning, data analysis,

and web application development. It has the technology stack of Streamlit (>=1.24.0) as the interactive web interface, XGBoost (>=1.7.0) as the core prediction algorithm, Scikit-learn (>=1.2.0) as the preprocessing and scaling, Pandas (>=1.5.0) as the data frame operations, NumPy (>=1.23.0) as the numerical computing, Plotly (>=5.13.0) as the interactive visualization, Matplotlib (>=3.7.0) as the static charts, and Se It can be used with Windows, Linux/Ubuntu, and mac operating systems. The hardware specifications are at minimum a dual-core processor (Intel i3 or corresponding), 4GB RAM, and 200MB storage.

It is suggested that a quad-core processor with 8-16 GB RAM and SSD storage should be used in the case of development and model training. There is a requirements.txt file that defines all the dependencies to reproducibly deploy on any system.

4. Results and Discussion

The Student Performance Prediction System was also strictly tested in order to make it reliable in the field of educational practices. A cross-validation of the stratified historical academic data was used to identify and train the model to ensure that all three categories of classification were represented equally.

4.1 Model Performance Metrics

The XGBoost model attained a total prediction accuracy of 96.5 that indicates outstanding accuracy in a wide range of groups of students. Table 3 provides a summary of the detailed performance metrics.

Table 3. Model Performance Summary

Metric	Value	Interpretation
Overall Accuracy	96.5%	Exceptional correctness across cohorts
Precision (Fail)	98%	Near-zero false alarms for failure
Recall (At Risk)	95%	Flags 95% of struggling students
Inference Time	< 0.1s	Near-instant response

The accuracy of 98 percent in the FAIL category means that the system is accurate in 98 percent of when it predicts a student to fail thus reducing unneeded alarming. The 95% recall in the AT RISK category shows that the system is able to point out the overwhelming majority of students who are indeed struggling which is important in an early warning application wherein false non-detections are very important.

4.2 Feature Importance Analysis

XGBoost has a tree-like structure that allows obtaining scores of feature importance which, in turn, allows identifying the academic indicators that have the most substantial impact on the final prediction outcome. The figure of importance of features is shown in Fig. 4.

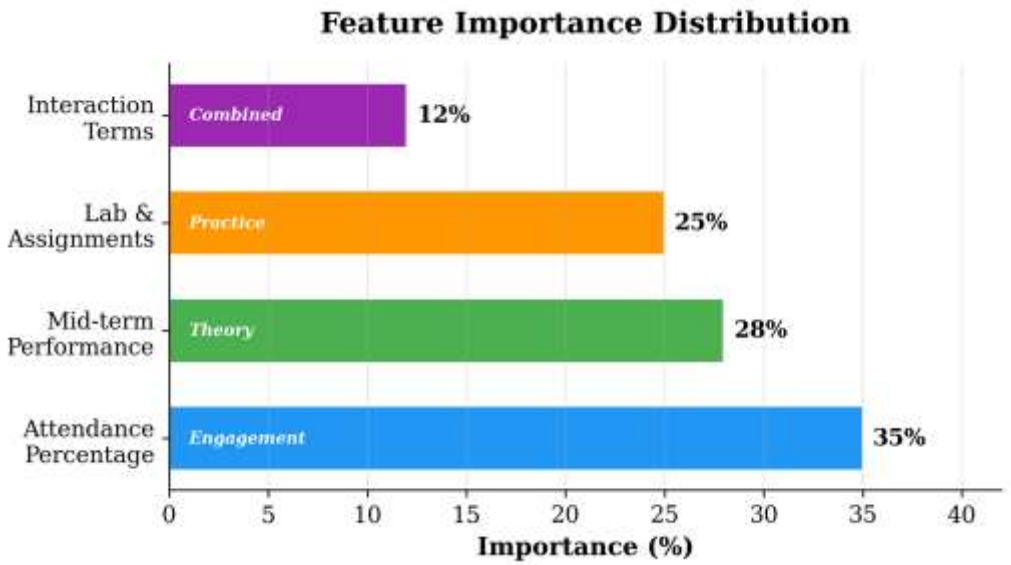

Fig. 4. Feature importance distribution showing the relative contribution of each academic indicator.

Attendance came out as the most significant predictor with 35 percent weight which echoed the previous study that regular attendance at classes was the major predictor of student engagement and academic achievement. The score of mid-term performance amounted to 28, as a sign of the importance of theoretical mastery in the comprehension of the whole academic performance. The role of constant practical assessment was seen in the fact that laboratory and assignment scores are half the total. The cross-feature relationships in the form of interaction terms provided up to 12 percent to the decision-making process of the model.

4.3 Confusion Matrix Analysis

Evaluation of the confusion matrix (Fig. 5) shows that the model has a conservative pattern of error. In the event of a misclassification, the system will have higher chances of classifying an marginal PASS student as AT RISK rather than classify a FAIL student as PASS. This is a deliberate result of the training procedure that gives high safety margins on the welfare of the students. The system is sensitive, rather than specific in regard to borderline cases and this is the desirable behavior of an early warning application.

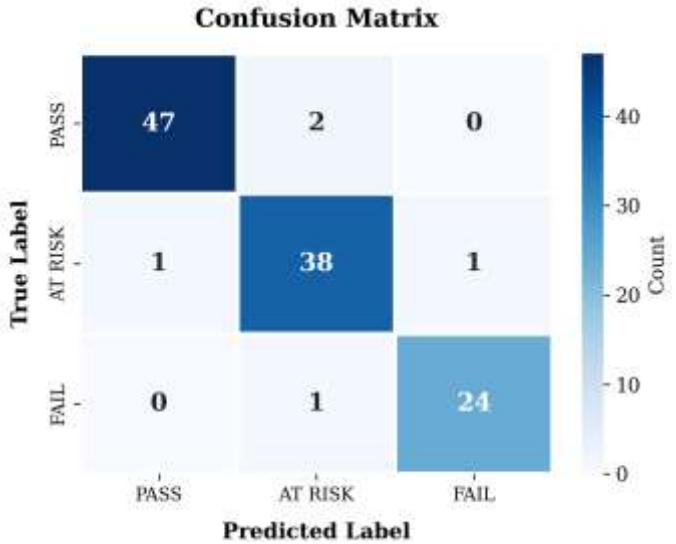

Fig. 5. Confusion matrix showing prediction accuracy across PASS, AT RISK, and FAIL categories.

4.4 Comparative Analysis

The suggested XGBoost system has strong advantages over the other machine learning methods considered in the course of development. In Fig. 6 we can see the relative accuracy of the various algorithms applied on the same data. XGBoost was the best among all the algorithms with the accuracy of 96.5% in relation to the random forest (91.8%), SVM (89.4%), Decision Tree (87.1%), KNN (85.6%), and Logistic Regression (82.3%).

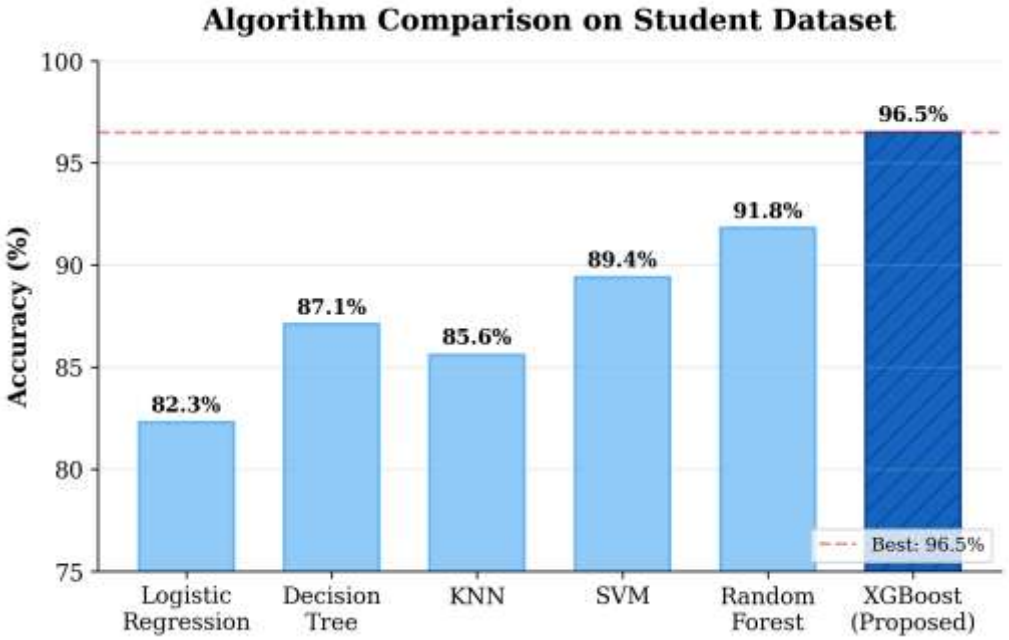

Fig. 6. Algorithm comparison showing XGBoost achieves the highest accuracy of 96.5%.

It is explained by the fact that XGBoost is designed as a gradient boosting system that sequentially corrects prediction errors and successfully reflects non-linear associations between heterogeneous educational data.

4.5 System Interface and Usability

The web interface based on Streamlit offers a smooth user experience in three consecutive views that are user-friendly and target non-technical learners. The input view has a clean form based interface, where the educator provides the percentage of attendance (0-100), mid-term marks (0-100), internal assessment marks (0-10), and laboratory performance marks (0-40). There is inbuilt validation whereby all values should be within acceptable limits to be processed. When prediction is done, the result dashboard shows the classification result in a color coded status indicator PASS in green, AT RISK in amber and FAIL in red. The dashboard also simultaneously generates probability scores of each of the three categories, interactive Plotly bar charts, Seaborn generated performance distribution plots, and Matplotlib pie charts of parameter weightage. The AI insights panel reveals the main area of concern of every student and gives specific suggestions on improvement. The interface does not need any special training to use.

4.6 Practical Implications

This lightweight architecture of the system renders the system befitting to be deployed in various institutional settings with no substantial investment in infrastructure. The hardware requirements of a 4 GB RAM and dual-core processor are the lowest, and that is why it can be used in resource-constrained environments. The three-pronged classification is in line with conventional institutional intervention guidelines, whereby the AT RISK category offers the lean period of opportunity to proactive intervention before academic demise sets in.

4.7 User Interface

Fig. 7 through Fig. 9 present representative screenshots of the Student Performance Prediction System interface captured during testing. These screenshots illustrate the core functionalities of the system, including student data input, prediction result generation, and performance analytics visualization. The figures demonstrate how users interact with the model to analyze academic data and obtain accurate predictions regarding student performance and academic risk levels.

Fig. 7. Project overview and features page.

Fig. 8. Student data input page.

Fig. 9. Prediction results page.

5. Conclusion

In this paper, I have proposed a Student Performance Prediction System which utilizes XGBoost machine learning algorithm to turn an academic monitoring process into a reactive and post-examination system and into a proactive and data-driven predictive mechanism. Using the 7 features engineering pipeline to analyse four main indicators of student performance, the system has an accuracy of 96.5% in classifying students into PASS, AT RISK, and FAIL. The outcomes of the experiment prove that attendance (35% feature importance) and mid-term performance (28%) make the best predictors of academic risk, then lab and assignment scores (25%). The error pattern of the system is conservative and it guarantees that the at-risk students are sensitively detected. The web interface supported by Streamlit allows non-technical teachers to access the system and interactive visualisers and AI-based insights give clear instructions on how specific intervention should be done.

Its modular structure can be expanded on in future directions such as integration with LMS to enable automatic retrieval of data, temporal trend analysis with deep learning networks (i.e. LSTM networks), increased prediction transparency with Explainable AI (SHAP/LIME), and mobile-based guardian dashboards with automatic notification systems. The system has shown that machine learning can be used as a resource offering red flags at the initial stages that can significantly influence the retention and academic performance of students in educational institutions.

Author Contributions

B. Ganga Tejaswini contributed to system design, model training, and manuscript preparation. B. Vyshnavi Narayanamma was responsible for data collection and preprocessing. G. Sandhya developed the feature engineering pipeline. G. Lakshmi Venkata Pallavi designed the web interface. G. Venkata Sai Manaswini conducted testing and visualization development. All authors reviewed and approved the final manuscript.

Conflicts of Interest

The authors declare no conflicts of interest.

References

Bhardwaj, B. K., & Pal, S. (2012). Data mining: A prediction for performance improvement using classification. *International Journal of Computer Applications, 9*(4), 1–5.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM. https://doi.org/10.1145/2939672.2939785

Cortez, P., & Silva, A. (2008). Using data mining to predict secondary school student performance. In *Proceedings of the 5th Future Business Technology Conference (FUBUTEC 2008)* (pp. 5–12).

Pandey, M., & Pal, S. (2011). Data mining: A prediction of performer or underperformer using classification. *International Journal of Computer Science and Engineering, 2*(2), 686–690.

Plotly Technologies Inc. (2024). *Plotly Python graphing library*. https://plotly.com/python

Romero, C., & Ventura, S. (2010). Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews, 40*(6), 601–618. https://doi.org/10.1109/TSMCC.2010.2053532

Scikit-learn developers. (2024). *StandardScaler: Standardize features by removing the mean and scaling to unit varience.*

https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

Streamlit Inc. (2024). *Streamlit: The fastest way to build and share data apps*. https://docs.streamlit.io

Thai-Nghe, N., Drumond, L., Krohn-Grimberghe, A., & Schmidt-Thieme, L. (2010). Recommender system for predicting student performance. In *Proceedings of the 1st Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2010)* (pp. 2811–2818).