A Comparative Machine Learning Approach for Predicting Student Academic Performance Using Ensemble and Classification Models Processing

DASARI AKSHAYA LAKSHMI

PG Scholar. Department of MCA, DNR College, Bhimavaram, Andhra Pradesh

V.SARALA

(Assistant Professor), Master of Computer Applications, DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

Student performance prediction has become a crucial application of machine learning in the domain of educational data mining. With the increasing availability of academic datasets, institutions can leverage predictive analytics to identify students at risk of poor academic outcomes and take proactive measures. This project presents a web-based intelligent system developed using Django that predicts student performance using multiple machine learning algorithms and compares their effectiveness.

The system utilizes a dataset containing various attributes influencing student performance, including demographic, social, and academic factors such as age, study time, parental occupation, health status, and past academic scores. Initially, the dataset undergoes preprocessing, where categorical data is converted into numerical form using label encoding. Missing values are handled efficiently, and feature scaling is performed using standardization to ensure uniformity across all attributes. The preprocessed dataset is divided into training and testing subsets. Several machine learning algorithms are applied, including K-Nearest Neighbors (KNN), Random Forest, Support Vector Machine (SVM), Gradient Boosting, Logistic Regression, and XGBoost. Each model is trained and evaluated using performance metrics such as accuracy, precision, recall, and F1-score. These metrics provide a comprehensive understanding of model performance.

The system also integrates a web interface where users can input student details and receive predictions about academic performance, categorized into outcomes such as “Pass” or “Dropout.” Based on predictions, the system provides recommendations, especially for students identified as at risk. Additionally, graphical visualizations such as pie charts and bar graphs are generated to present insights into student performance trends and algorithm comparisons. Among the implemented models, ensemble methods like Random Forest and Gradient Boosting generally demonstrate better predictive performance due to their ability to handle complex relationships in data. The inclusion of XGBoost further enhances prediction accuracy by leveraging gradient boosting techniques with optimized performance. This project not only highlights the importance of machine learning in education but also provides a practical solution for early identification of underperforming students. The system can assist educators and administrators in making data-driven decisions to improve student outcomes. Future

enhancements may include real-time data integration, deep learning models, and personalized learning recommendations.

Keywords: Student Performance Prediction, Machine Learning, Random Forest, XGBoost, Gradient Boosting, Classification Algorithms, Educational Data Mining, Predictive Analytics, Django Framework, Data Preprocessing

I. INTRODUCTION

Education plays a vital role in shaping the future of individuals and society. Monitoring and improving student performance is a key responsibility of educational institutions. Traditional methods of evaluating student performance often rely on manual analysis, which can be time-consuming and less effective in identifying underlying patterns. With advancements in machine learning, it is now possible to automate this process and generate accurate predictions based on historical data. This project focuses on developing a machine learning-based system to predict student academic performance. The system is designed using the Django framework, which provides a robust and scalable platform for deploying web-based applications. By integrating machine learning models into a web interface, the system allows users to interactively input student data and obtain predictions in real time. The dataset used in this project includes various attributes that influence academic performance. These attributes are categorized into personal, academic, and socio-economic factors. Examples include gender, age, parental occupation, study time, failures, and previous grades. Such a diverse set of features enables the system to capture complex relationships affecting student outcomes. Data preprocessing is a critical step in this project. Since the dataset contains categorical variables, label encoding is applied to convert them into numerical values. Standardization is performed to normalize feature values, ensuring that all attributes contribute equally to model training. The dataset is then split into training and testing sets to evaluate model performance effectively.

Multiple machine learning algorithms are implemented to compare their effectiveness. KNN is used for its simplicity and instance-based learning approach. Random Forest and Gradient Boosting are ensemble methods known for their high accuracy and robustness. SVM is utilized for its capability to handle high-dimensional data, while Logistic Regression serves as a baseline model. XGBoost is included for its advanced boosting capabilities and efficiency. The system evaluates each model using performance metrics such as accuracy, precision, recall, and F1-score. These metrics provide a comprehensive evaluation, ensuring that the models are not only accurate but also reliable in predicting different classes. A user-friendly web interface is developed to facilitate interaction. Users can input student details, and the system predicts performance outcomes. Additionally, graphical representations such as pie charts and bar graphs are generated to visualize results and model comparisons. Overall, this project demonstrates how machine learning can be effectively applied in education to enhance decision-making and improve student outcomes.

LITERATURE SURVEY (WITH EXISTING METHODS)

Student performance prediction has been widely studied in the field of educational data mining. Various machine learning techniques have been explored to analyze academic data and identify patterns influencing student success. One of the earliest approaches involved statistical methods such as linear regression, which aimed to model relationships between student attributes and performance outcomes. While these methods provided basic insights, they were limited in handling complex and non-linear relationships. With the advancement of machine learning, classification algorithms such as Decision Trees and K-Nearest Neighbors (KNN) gained popularity. Decision Trees are easy to interpret and can handle both categorical and numerical data. KNN, on the other hand, is a simple yet effective algorithm that classifies data based on similarity measures. However, these methods often suffer from overfitting and scalability issues. Support Vector Machines (SVM) have also been widely used for student performance prediction. SVM is effective in handling high-dimensional data and can model complex decision boundaries. However, it requires careful parameter tuning and may not perform well with large datasets.

Ensemble learning methods such as Random Forest and Gradient Boosting have shown significant improvements in prediction accuracy. Random Forest combines multiple decision trees to reduce overfitting and improve generalization. Gradient Boosting builds models sequentially, correcting errors from previous models. These methods are highly effective in capturing complex relationships in data. More recently, advanced techniques like XGBoost have gained popularity due to their efficiency and scalability. XGBoost is an optimized implementation of gradient boosting that includes regularization techniques to prevent overfitting. It has been widely used in various prediction tasks, including educational data analysis. Deep learning approaches, such as Artificial Neural Networks (ANNs), have also been explored for student performance prediction. These models can capture intricate patterns in data but require large datasets and significant computational resources. Existing studies have demonstrated that combining multiple algorithms and comparing their performance can lead to better insights. Visualization techniques such as confusion matrices, bar graphs, and pie charts are commonly used to present results. The current project builds upon these existing methods by implementing multiple machine learning algorithms within a single system and comparing their performance. By integrating these models into a web-based application, the project provides a practical solution for real-world use.

II. EXISTING SYSTEM

The existing systems for student performance evaluation are primarily based on traditional methods such as manual assessment, statistical analysis, and basic data processing techniques. Educational institutions often rely on teachers' observations, exam scores, and attendance records to evaluate student performance. While these methods provide useful insights, they are limited in their ability to analyze large datasets and

identify hidden patterns. Some existing systems use basic statistical models such as linear regression to predict student outcomes. Although these models are simple to implement, they are not capable of capturing complex relationships between multiple influencing factors. As a result, their prediction accuracy is often limited. In recent years, machine learning-based systems have been introduced to improve prediction accuracy. However, many of these systems focus on a single algorithm, which may not provide the best results for all types of data. Additionally, some systems lack proper data preprocessing, which can significantly affect model performance. Another limitation of existing systems is the lack of user-friendly interfaces. Many solutions are developed as standalone applications without web integration, making them less accessible to users. Furthermore, visualization features are often limited, reducing the ability to interpret results effectively. Data handling is also a challenge in existing systems. Issues such as missing values, inconsistent data formats, and unbalanced datasets are not always addressed properly. This can lead to inaccurate predictions and unreliable results.

Overall, the existing systems are limited by their reliance on traditional methods, lack of algorithm diversity, insufficient preprocessing, and poor user interaction. These limitations highlight the need for an improved system that integrates multiple machine learning techniques, proper data handling, and an interactive web-based interface.

III. PROPOSED METHOD

The proposed system aims to develop an intelligent and automated student performance prediction platform using multiple machine learning algorithms integrated into a web-based environment. Unlike traditional systems that rely on single models or manual evaluation, this system combines various classification techniques to improve prediction accuracy and reliability. The system begins with data preprocessing, where raw student data is cleaned and transformed. Categorical attributes such as gender, parental occupation, and educational support are converted into numerical values using label encoding. Missing values are handled effectively, and feature scaling is applied using standardization to ensure uniform data distribution. These preprocessing steps significantly enhance model performance. The core of the system involves implementing multiple machine learning algorithms, including K-Nearest Neighbors (KNN), Random Forest, Support Vector Machine (SVM), Gradient Boosting, Logistic Regression, and XGBoost. Each algorithm is trained using the same dataset, allowing for a comparative analysis of their performance. Evaluation metrics such as accuracy, precision, recall, and F1-score are calculated to determine the most effective model.

The system provides a user-friendly interface where users can input student details and obtain predictions in real time. The output categorizes student performance into predefined classes such as “Pass” or “Dropout.” Additionally, the system provides recommendations for improvement, especially for students at risk. Visualization features are integrated to enhance interpretability. Graphs such as pie charts and bar charts display performance distribution and algorithm comparisons. This helps educators understand trends and make informed decisions. The proposed system leverages ensemble learning techniques, which have been shown to improve prediction accuracy by combining

multiple models . Overall, the system provides a scalable, efficient, and accurate solution for predicting student performance.

IV. IMPLEMENTATION

The implementation of the student performance prediction system is carried out using Python, Django framework, and machine learning libraries such as Scikit-learn and XGBoost. The system is designed as a web-based application to ensure accessibility and ease of use. The first step in implementation involves loading the dataset using the Pandas library. The dataset contains various attributes related to student demographics, academic performance, and social factors. Data preprocessing is performed to clean and prepare the dataset for model training. Categorical variables are converted into numerical format using label encoding, ensuring compatibility with machine learning algorithms. Missing values are replaced with default values to avoid inconsistencies. Feature scaling is applied using the StandardScaler to normalize the dataset. This ensures that all features contribute equally during model training. The dataset is then shuffled and split into training and testing sets using the train_test_split method. This step is essential for evaluating model performance on unseen data.

Multiple machine learning models are implemented in the system. KNN is used for its simplicity and effectiveness in classification tasks. Random Forest is employed for its ability to handle large datasets and reduce overfitting. SVM is used for its capability to create optimal decision boundaries. Gradient Boosting and XGBoost are implemented for their high accuracy and efficiency in handling complex data relationships. Logistic Regression is used as a baseline model for comparison. Each model is trained using the training dataset and evaluated on the testing dataset. Performance metrics such as accuracy, precision, recall, and F1-score are calculated using Scikit-learn functions. These metrics provide a comprehensive evaluation of each model's effectiveness. The Django framework is used to create the web interface. Views are defined to handle user requests, process input data, and display results. Users can enter student details through a form, and the system processes this data to generate predictions using the trained model. The prediction results are displayed along with recommendations. Visualization is implemented using Matplotlib and Seaborn libraries. Graphs are generated dynamically and displayed on the web interface using base64 encoding. These visualizations help users understand performance trends and model comparisons. The system also includes functionalities for dataset upload, admin login, and result visualization. Overall, the implementation ensures a seamless integration of machine learning and web technologies to provide an efficient and user-friendly solution.

V. ALGORITHMS

The system utilizes several machine learning algorithms to predict student performance. Each algorithm has unique characteristics that contribute to the overall effectiveness of the system. K-Nearest Neighbors (KNN) is a simple and instance-based learning algorithm that classifies data based on the proximity of neighboring data points. It is effective for small datasets but may face scalability issues. Random Forest is an ensemble

learning method that combines multiple decision trees to improve prediction accuracy and reduce overfitting. It is widely used due to its robustness and ability to handle complex data structures. Support Vector Machine (SVM) is a powerful classification algorithm that finds the optimal hyperplane to separate data into different classes. It performs well in high-dimensional spaces but requires careful parameter tuning. Gradient Boosting is another ensemble technique that builds models sequentially by correcting errors from previous models. It is highly effective for improving prediction accuracy.

Logistic Regression is a statistical method used for binary classification problems. It provides a baseline for comparison with more complex algorithms. XGBoost (Extreme Gradient Boosting) is an advanced implementation of gradient boosting that includes regularization techniques to prevent overfitting and improve performance. It is known for its efficiency and scalability. Research studies have shown that ensemble methods such as Random Forest and XGBoost outperform traditional models in student performance prediction tasks . By combining multiple algorithms, the system ensures better accuracy and reliability in predictions.

VI. SYSTEM DESIGN

The system design follows a modular and layered architecture to ensure scalability, maintainability, and efficiency. The architecture consists of three main layers: presentation layer, application layer, and data layer. The presentation layer is responsible for user interaction. It is developed using HTML templates integrated with the Django framework. Users can access different functionalities such as dataset upload, prediction input, and result visualization through a web interface. Forms are used to collect input data from users, which is then processed by the backend. The application layer handles the core logic of the system. It includes Django views that process user requests and interact with machine learning models. When a user submits input data, the system preprocesses the data using label encoding and feature scaling. The processed data is then passed to the trained machine learning model for prediction.

The machine learning module is a crucial component of the system. It includes multiple algorithms such as KNN, Random Forest, SVM, Gradient Boosting, Logistic Regression, and XGBoost. These models are trained using historical student data and stored for future predictions. The system also calculates performance metrics for each model and compares their effectiveness. The data layer manages the dataset and model storage. The dataset is stored in CSV format and loaded using Pandas. Processed data and trained models are handled efficiently to ensure quick access during prediction. The system also includes visualization components. Graphs are generated using Matplotlib and displayed on the web interface. These visualizations provide insights into student performance distribution and algorithm comparison. From a design perspective, the system follows best practices such as modular coding, separation of concerns, and reusable components. This ensures that the system can be easily extended with additional features such as real-time data processing or advanced machine learning models. The design also considers scalability and performance. Efficient data handling and optimized algorithms ensure that the system can handle large datasets. Security features such as admin authentication are

implemented to protect sensitive data. Overall, the system design provides a robust framework for integrating machine learning with web technologies to deliver accurate and efficient student performance predictions.

SYSTEM DESIGN IMAGES

In this project as per your instructions we have designed application to display dataset values and then train various machine learning algorithms like SVM, Random Forest, XGBOOST, KNN, Logistic Regression and Gradient Boosting. Each algorithm performance is evaluated in terms of user accuracy, precision, recall and FSCORE.

All algorithm manages to give an accuracy of over 90 to 99% and among all algorithms XGBOOST stand winner with an accuracy of over 98%.

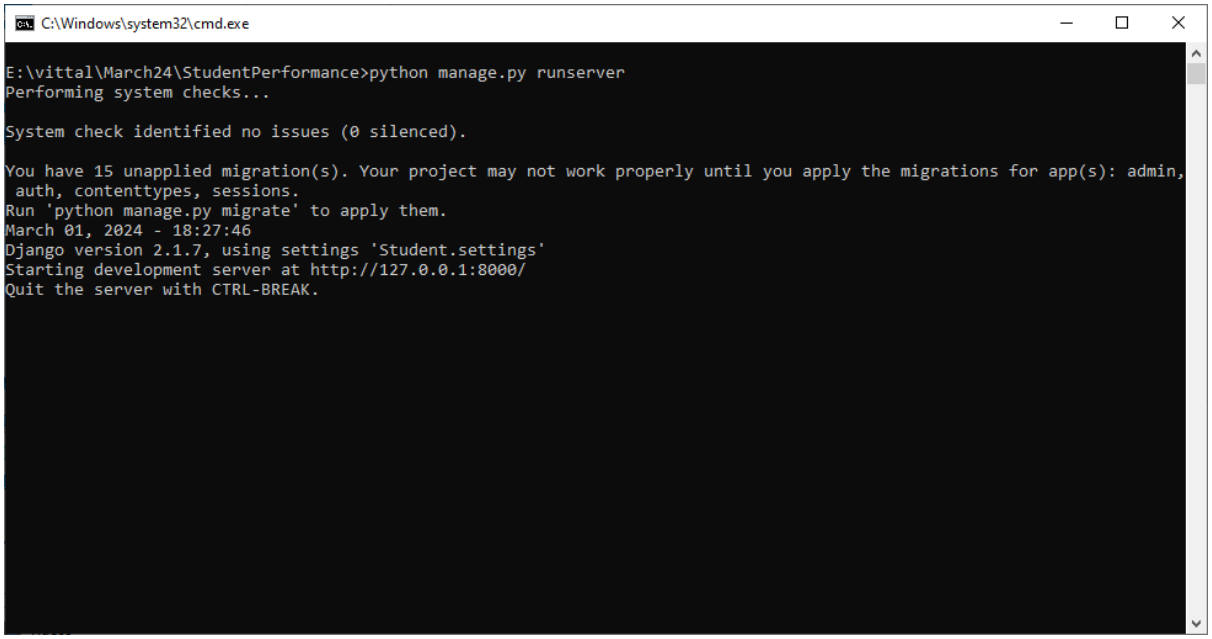
If student performance poor then system will display alert message to Focus and work hard.

To implement this project we have designed following modules

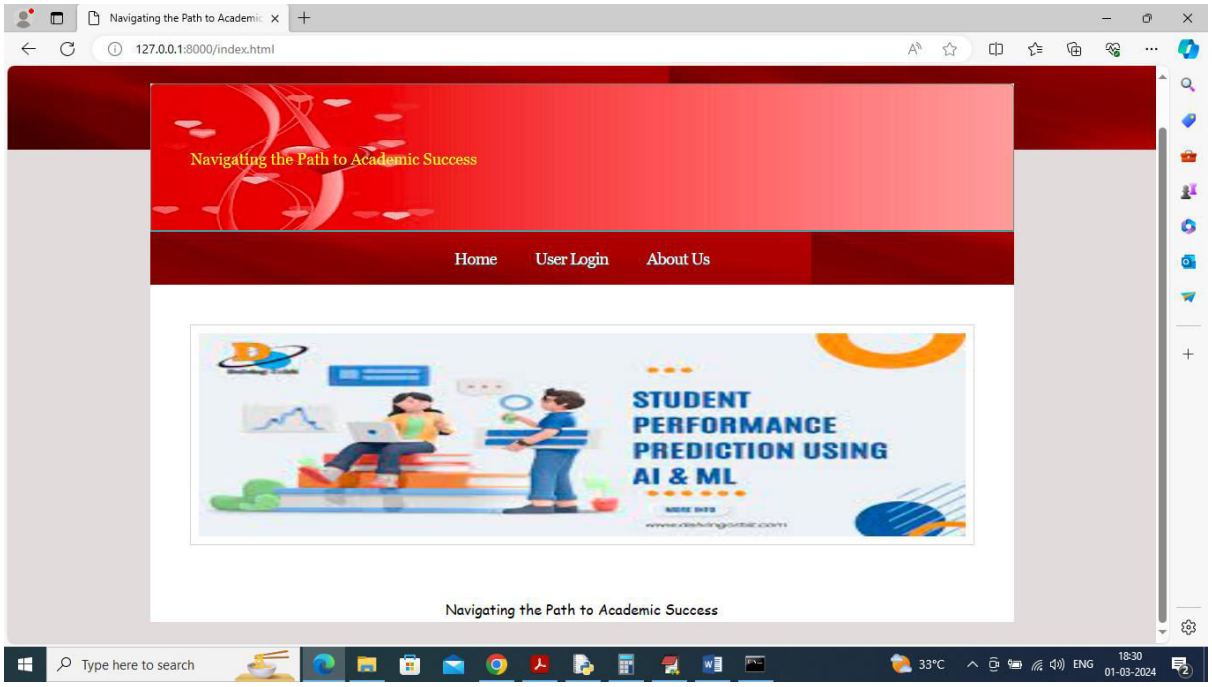
1) User Login: user can login to system using username and password as ‘admin and admin’.
2) Load & Process Dataset: user can load dataset and then display and process dataset values
3) Train ML algorithms: this module will train all algorithms and then display training result in table and graph format
4) Predict Performance: user will input his academic details and then ML algorithm will predict his performance
5) Graph Analysis: will plot PIE chart of all student performance

SCREEN SHOTS

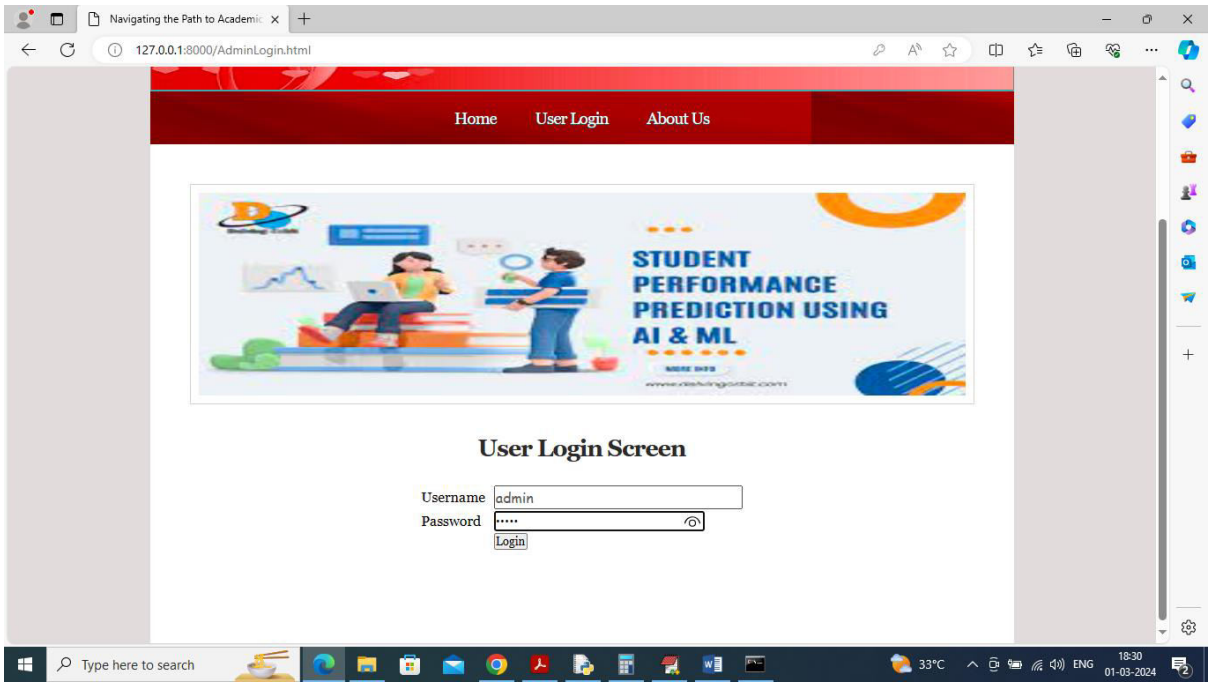
To run project double click on run.bat file to start python server and get below page



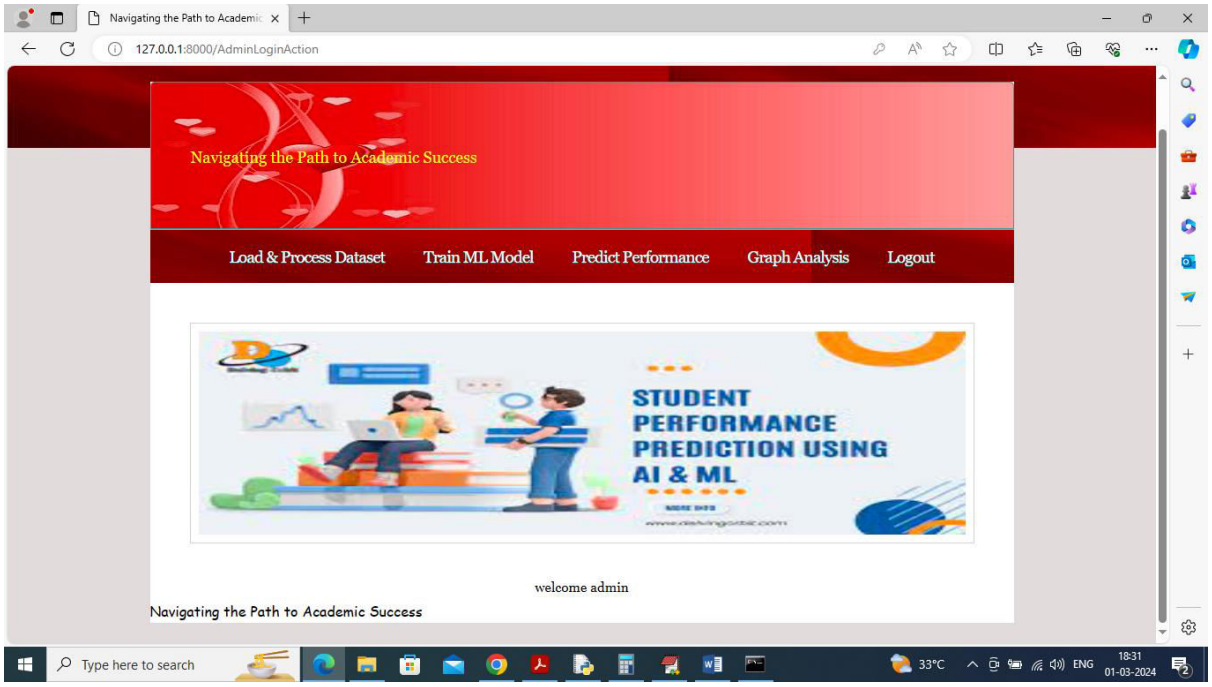
In above screen python server started and now open browser and enter URL as http://127.0.0.1:8000/index.html and press enter key to get below page



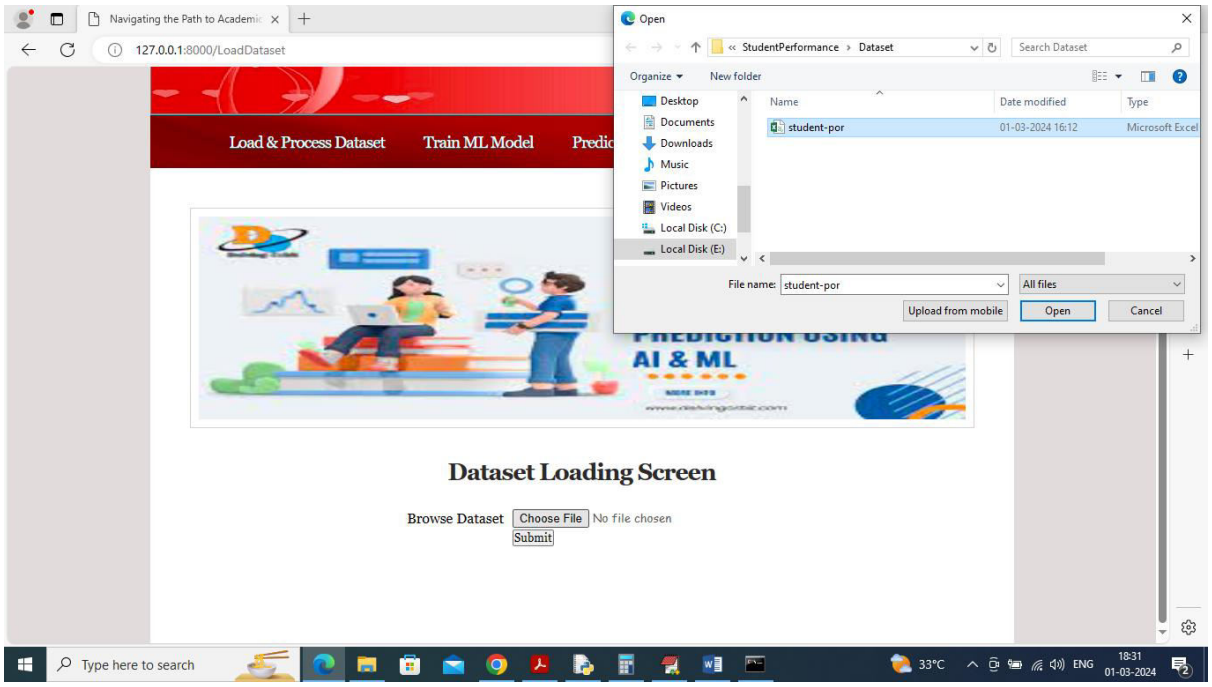
In above screen click on 'User Login' link to get below page



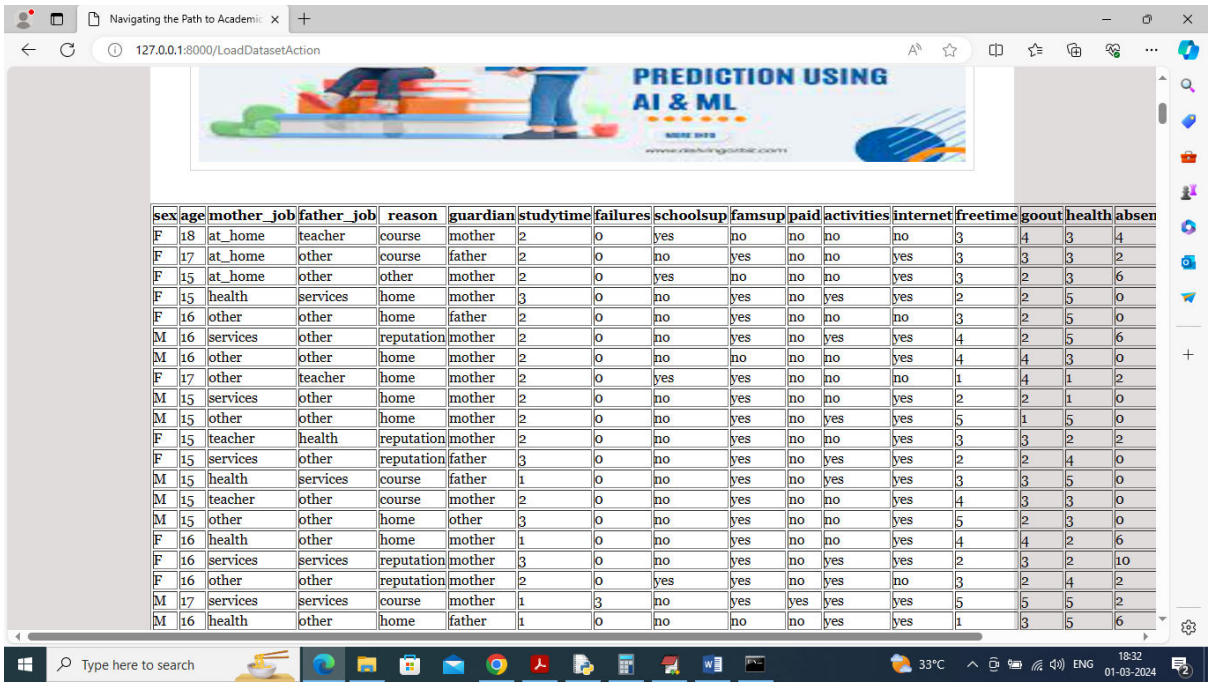
In above screen user is login and after login will get below page



In above screen user can click on ‘Load & Process Dataset’ link to get below page



In above screen select and load dataset file and this dataset file available inside 'Dataset' folder and then click on 'Open' and 'Submit' button to get below page

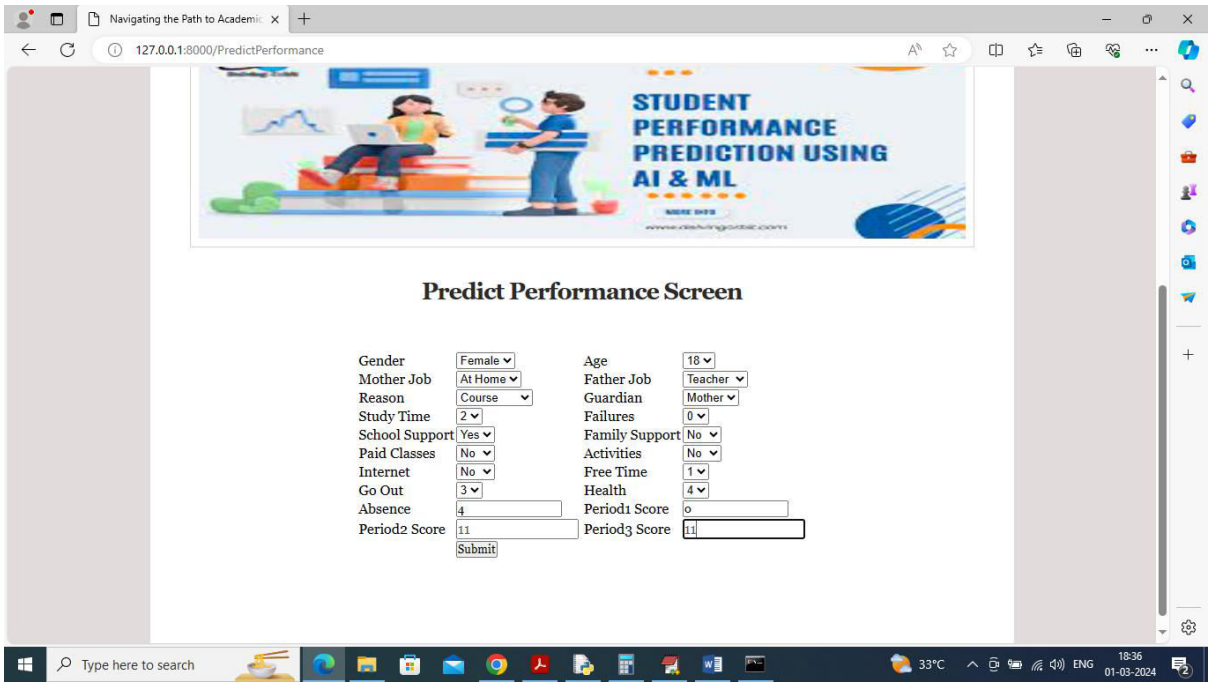


sex	age	mother_job	father_job	reason	guardian	studytime	failures	schoolsup	famsup	paid	activities	internet	freetime	gout	health	absen
F	18	at_home	teacher	course	mother	2	0	yes	no	no	no	no	3	4	3	4
F	17	at_home	other	course	father	2	0	no	yes	no	no	yes	3	3	3	2
F	15	at_home	other	other	mother	2	0	yes	no	no	no	yes	3	2	3	6
F	15	health	services	home	mother	3	0	no	yes	no	yes	yes	2	2	5	0
F	16	other	other	home	father	2	0	no	yes	no	no	no	3	2	5	0
M	16	services	other	reputation	mother	2	0	no	yes	no	yes	yes	4	2	5	6
M	16	other	other	home	mother	2	0	no	no	no	no	yes	4	4	3	0
F	17	other	teacher	home	mother	2	0	yes	yes	no	no	no	1	4	1	2
M	15	services	other	home	mother	2	0	no	yes	no	no	yes	2	2	1	0
M	15	other	other	home	mother	2	0	no	yes	no	yes	yes	5	1	5	0
F	15	teacher	health	reputation	mother	2	0	no	yes	no	no	yes	3	3	2	2
F	15	services	other	reputation	father	3	0	no	yes	no	yes	yes	2	2	4	0
M	15	health	services	course	father	1	0	no	yes	no	yes	yes	3	3	5	0
M	15	teacher	other	course	mother	2	0	no	yes	no	no	yes	4	3	3	0
M	15	other	other	home	other	3	0	no	yes	no	no	yes	5	2	3	0
F	16	health	other	home	mother	1	0	no	yes	no	no	yes	4	4	2	6
F	16	services	services	reputation	mother	3	0	no	yes	no	yes	yes	2	3	2	10
F	16	other	other	reputation	mother	2	0	yes	yes	no	yes	no	3	2	4	2
M	17	services	services	course	mother	1	3	no	yes	yes	yes	yes	5	5	5	2
M	16	health	other	home	father	1	0	no	no	no	yes	yes	1	3	5	6

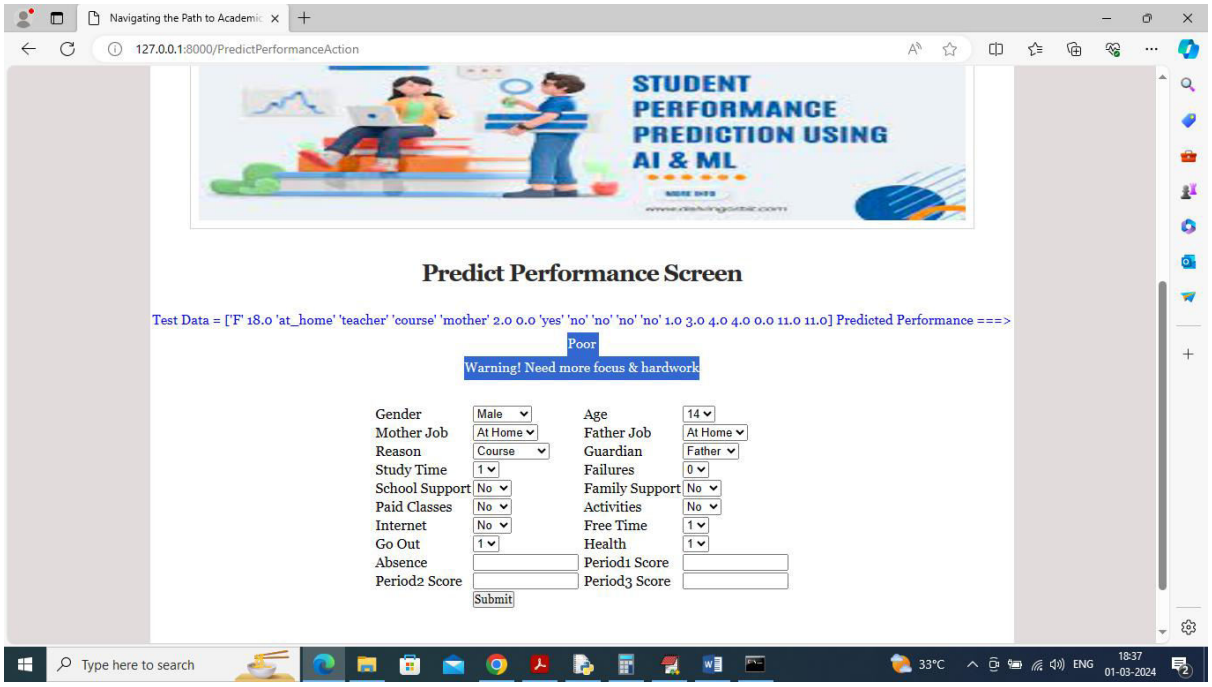
In above screen dataset loaded and can see all columns and its values and now click on 'Train ML Algorithm' link to train all algorithms and get below page

Algorithm Name	Accuracy	Precision	Recall	FSCORE
KNN	73.07692307692307	82.01754385964911	52.615590567397795	57.2286117317573
Random Forest	95.38461538461539	96.4176245210728	92.57763655354016	94.34368869499018
SVM	96.92307692307692	97.49019607843138	95.44698942289304	96.43255676254242
Gradient Boosting	99.23076923076923	99.60317460317461	98.41269841269842	98.9873910715155
Logistic Regression	95.38461538461539	96.75095785440612	92.27238624828985	94.29864253393664
XGBoost	98.46153846153847	98.01109198699561	98.01109198699561	98.01109198699561

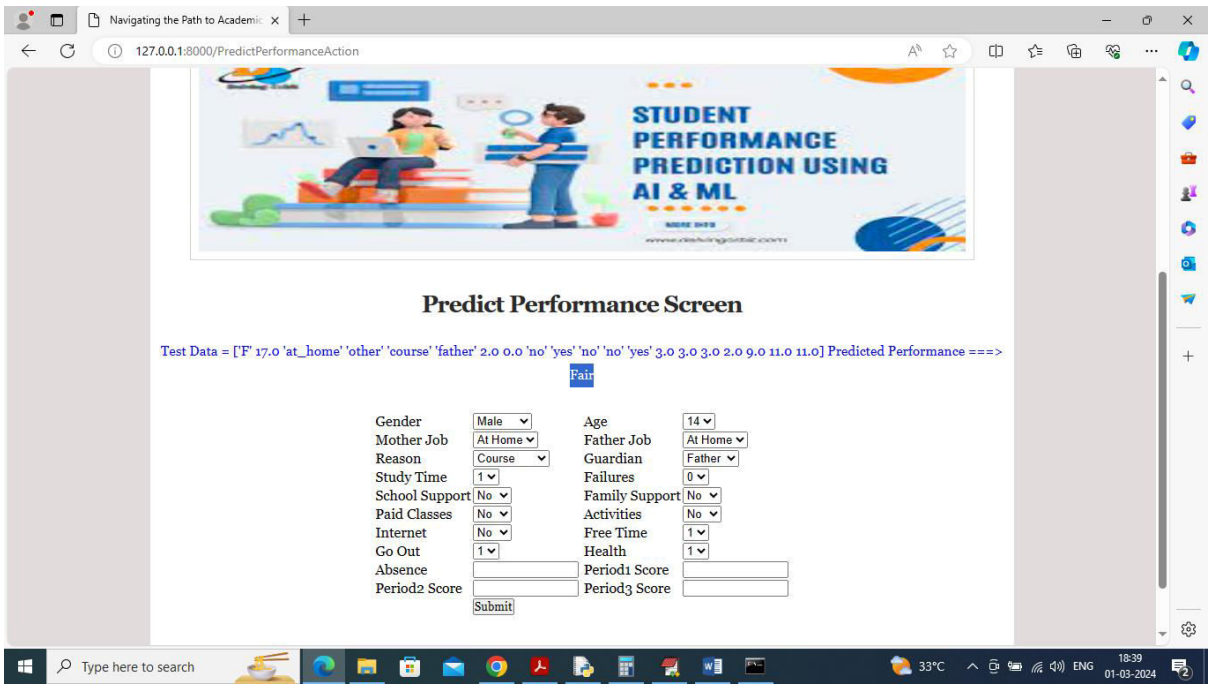
In above screen can see each algorithm performance in tabular format and in graph format. In graph x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars and in all algorithms Gradient boosting and XGBOOST got high accuracy and now click on 'Predict Performance' link to get below page



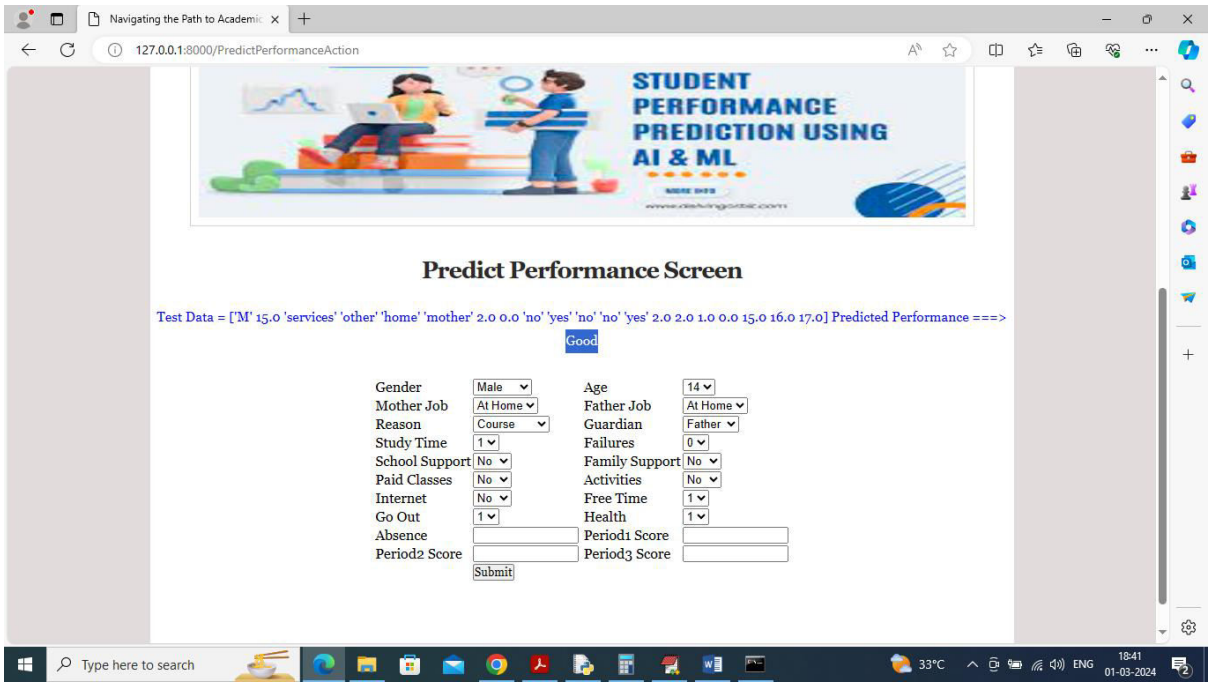
In above screen user will enter and select academic details and then click on 'Submit' button to get below output



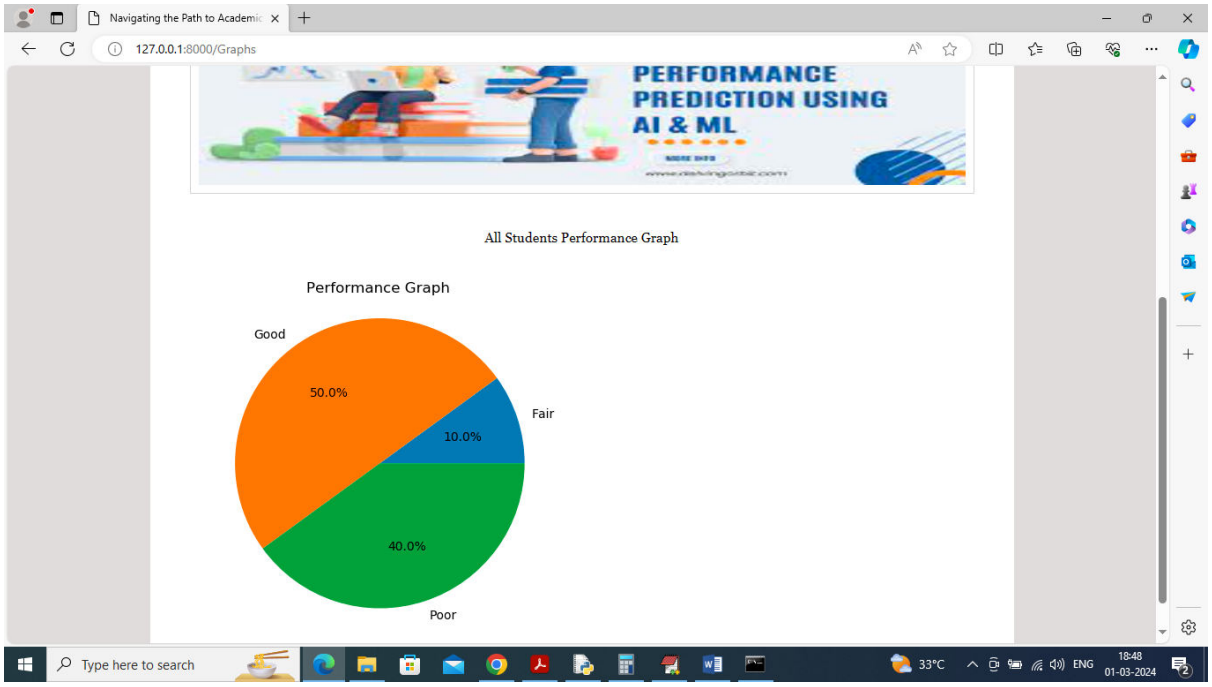
In above screen in blue colour can see user academic data and then can predicted performance as 'Poor' with alert message to improve. Similarly you can input any details and get performance predicted. Below is another output



In above screen predicted performance is 'Fair'.



In above screen predicted performance is ‘Good’.



In above pie chart graph can see overall performance of all students

VII. CONCLUSION

The student performance prediction system demonstrates the effective application of machine learning techniques in the field of education. By leveraging historical student data and advanced algorithms, the system provides accurate predictions that can help identify students at risk of poor academic performance. The implementation of multiple machine learning models allows for a comprehensive comparison of their performance. Ensemble methods such as Random Forest and XGBoost have shown superior accuracy due to their ability to handle complex data patterns. The use of evaluation metrics such as accuracy, precision, recall, and F1-score ensures a reliable assessment of model performance. The integration of the system into a web-based platform using Django enhances accessibility and usability. Users can easily input student data and obtain predictions in real time. Visualization features further improve the interpretability of results, enabling educators to make informed decisions. This system addresses the limitations of traditional methods by providing a data-driven approach to student performance evaluation. It enables early identification of at-risk students, allowing timely intervention and support. Such predictive systems can significantly improve educational outcomes and reduce dropout rates.

Recent research highlights the importance of machine learning in improving academic performance prediction and decision-making in educational institutions . The proposed system aligns with these advancements and provides a practical implementation. Future enhancements may include the integration of deep learning models, real-time data analytics, and personalized recommendations. Additionally, incorporating more diverse datasets can further improve prediction accuracy. In conclusion, the system provides a scalable, efficient, and intelligent solution for student performance prediction, contributing to the advancement of educational data mining and smart learning systems.

REFERENCES

1. Jayasree R., Selvakumari S., “Design of a Prediction Model to Predict Students’ Performance Using Educational Data Mining,” 2023
2. Tiwari M., Jain N., “Student Performance Prediction Using Machine Learning Algorithms,” 2024
3. Ahmed E., “Student Performance Prediction Using Machine Learning Algorithms,” 2024
4. Rahul & Katarya, “Systematic Review on Predicting Student Performance,” 2024
5. Ahmed W. et al., “Machine Learning-Based Academic Performance Prediction,” 2025
6. Tang B. et al., “Deep Ensemble Learning for Student Performance Prediction,” 2024
7. Hassan M. et al., “Supervised ML Models for Student Performance,” 2026

8. Yadav N., Deshmukh S., “Prediction of Student Performance Using ML Techniques,” 2023
9. Duan C. et al., “Systematic Literature Review on Student Performance Prediction,” 2025
10. Sandeepa A., Mohottala S., “Evaluation of ML Models in Academic Prediction,” 2025
11. Wang Y. et al., “Graph-Based Ensemble ML for Student Prediction,” 2021
12. Jimenez A. et al., “Early Detection of At-Risk Students Using ML,” 2024
13. Kim B. et al., “GritNet: Deep Learning for Student Prediction,” 2018
14. ScienceDirect, “Classification and Prediction of Student Performance,” 2023
15. Atlantis Press, “Prediction of Student Performance Using ML: A Review,” 2023