JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

EARLY PREDICTING OF STUDENTS PERFORMANCE IN HIGHER EDUCATION

Chintala Venkata Keerthi[1] Nayanolla Nishitha2, N.S.R.K Prasad[3]

UG Student, Department of Information Technology, Guru Nanak Institutions Technical Campus, Hyderabad, Telangana, India

UG Student, Department of Information Technology, Guru Nanak Institutions Technical Campus, Hyderabad, Telangana, India
Assistant Professor, Department of Information Technology, Guru Nanak Institutions Technical Campus, Hyderabad, Telangana, India.

ABSTRACT

Assessing student learning performance stands as a pivotal aspect in evaluating the efficacy of educational institutions. Student performance serves as a quantitative measure to gauge the outcomes of the learning process. Within the realm of educational research, the field of Educational Data Mining (EDM) has burgeoned due to its potential in harnessing data-driven insights to optimize educational systems. EDM involves the development and application of methodologies to scrutinize data extracted from educational settings, facilitating a more comprehensive understanding of students and the improvement of their educational outcomes. Consequently, evaluating student learning outcomes remains imperative in the broader evaluation framework of educational institutions, with student performance serving as a crucial metric in addressing challenges pertinent to the learning process.

Keywords — ***Student learning performance, Educational institutions, Quantitative measure Learning outcomes, Educational Data Mining (EDM),Data-driven insights, Comprehensive understanding, Improvement, Evaluation framework, Challenges, Metric.***

I. INTRODUCTION

Education holds a prominent position in society, with information and communication technology (ICT) significantly influencing various research domains, particularly education. The recent COVID-19 pandemic has underscored the reliance on e-Learning platforms across many nations. Academic performance in higher education institutions is a key indicator of delivering quality education. Yet, comprehending the determinants of student performance in early education stages is intricate. While several effective tools address academic performance challenges, their applicability across educational contexts varies. Despite technological advancements facilitating student performance forecasting, gaps persist in

Analysing and enhancing accuracy using novel features and data mining methodologies. This study aims to employ clustering and classification techniques to discern the early-stage impact on GPA, alongside proposing a teaching management theory and data analysis algorithm grounded in clustering algorithms and educational big data. It seeks to establish a teaching quality evaluation index system utilizing one-dimensional data, facilitating teaching big data analysis. The research validates the functionality of the teaching management system and affirms the accuracy of data analysis results, furnishing objective insights for educational management personnel across hierarchical levels.

II. EXISTING SYSTEM

Evaluating educational systems hinges significantly on students' learning performance, pivotal for addressing learning process issues and gauging learning outcomes. The emergence of educational data mining (EDM) reflects the endeavour to leverage data knowledge in enhancing educational systems.

III.LITERATURE SURVEY

Title: Fog computing and its role in the internet of things.
Author: F. Bonomi, R. Milito, J. Zhu, and S. Addepalli
Year: 2012Description: Fog computing extends the Cloud Computing paradigm to the edge of the network, facilitating new applications and services. Its defining characteristics include low latency, location awareness, widespread geographical distribution, mobility, a large number of nodes, predominant wireless access, strong streaming and real-time application presence, and heterogeneity.
Title: Micro services scheduling model over heterogeneous cloud-edge environments as support for IoT applications.
Author: I.-D. Filip, F. Pop, C. Serbanescu, and C. Choi
Year: 2018Description: This model aims to enhance the utilization of non-general-purpose devices for computational tasks at reduced costs. It proposes a scheduling model for microservices over heterogeneous cloud-edge environments, employing a mathematical formulation. The model necessitates early risk analysis, prompting improvements in the Clouds simulation framework to accommodate such systems.
Title: Design and performance evaluation of containerized microservices on edge gateway in mobile IoT.
Author: A. S. Gaur, J. Budakoti, and C.-H. Lung
Year: 2018Description: The study addresses challenges in mobile IoT, such as seamless connectivity, efficient data transfer, and service management at the Edge Gateway. It advocates containerized virtualization solutions for managing and deploying microservices, enhancing connectivity in mobile IoT environments.
Title: Dyme: Dynamic microservice scheduling in edge computing enabled IoT.
Author: A. Samanta and J. Tang
Year: 2020Description: With the emergence of mobile edge computing (MEC), this study focuses on optimizing task execution in IoT applications considering varying network conditions. It aims to maximize energy efficiency and ensure fair Quality-of-Service (QoS) by dynamically scheduling tasks in edge platforms.
Title: Secure edge computing management based on independent micro services providers for gateway-centric IoT networks.
Author: W. Jin, R. Xu, T. You, Y.-G. Hong, and D. Kim
Year: 2020Description: Edge computing decentralizes computational capability to network edges, offering heterogeneous solutions closer to deployed sensors and actuators. The paper proposes secure edge computing management to address resource constraints, ensuring secure service provision from the network edge.
Title: Docker enabled virtualized nanos ervices for local IoT edge networks.
Author: J. Islam, E. Harjula, T. Kumar, P. Karhula, and M. Ylianttila
Year: 2019Description: This work explores Docker-enabled virtualized nanoservices for local IoT edge networks, focusing on aspects such as mobile communication, IoT, quality of service, reliability, security, and virtualization in telecommunication networks, particularly LTE and 5G.

IV. PROPOSED SYSTEM

This study offers an exhaustive assessment grounded in an authentic prototype implementation and performance evaluation. Within our configuration, an edge server assumes a dual function: serving as both an administrative controller for the IoT infrastructure and meeting the latency and privacy requirements of applications.

V.SYSTEM ARCHITECTURE

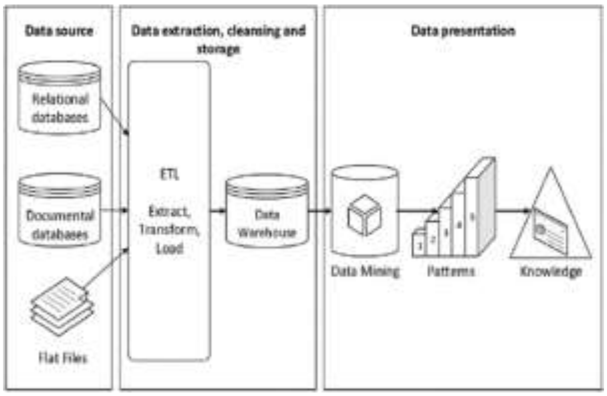


Fig: System Architecture Model

VI. METHODOLOGY

Our project operates under the assumption that the data owner is trustworthy, and only authorized users, vetted by the data owner, access the data. Communication between the data owner and users occurs over secure channels, leveraging established security protocols such as SSL and TLS.

Unlike traditional secure semantic searching schemes that rely on a "semi-honest server," our scheme withstands a more stringent security model. Here, the cloud server may attempt to manipulate search results and extract sensitive information but refrains from maliciously altering or deleting outsourced documents. Thus, our secure semantic scheme must ensure verifiability and confidentiality under this elevated security model.

User: This module encompasses the user interface, providing secure login functionality for all users. To establish a connection with the server, users must authenticate themselves by providing their username and password.
Pre-processing: Serving as the initial module, Data Users can register and log in. Upon logging in, Data Users gain access to a file search feature, enabling them to search for files by name. Additionally, users have the option to download files, which are presented in encrypted format.
Classifier: This module, the second in the project, caters to Data Owners who register and log in. Data Owners utilize this module to upload files into the database. Additionally, they can send requests to Data Users.
Training Data: Constituting the third module, Cloud Server login is facilitated here. Upon login, the Cloud Server gains visibility into all Data Owners' and users' information, as well as access to all stored data files.

VII.WORK FLOW

Logging and Utility Functions

1. Introduction:
This document outlines the implementation details of logging and utility functions in a Python project. Logging is essential for tracking and debugging errors, while utility functions provide various functionalities to support the project's operations.

2. Logging Setup:

The logging functionality is established using Python's built-in logging module.
A custom logger is created to handle logging operations throughout the project.
Logging is configured to output to a file with a timestamped filename in a designated logs directory.
The log format includes timestamp, line number, module name, log level, and message.
3. Custom Exception Handling:
A Custom Exception class is defined to handle exceptions in a customized manner.
The class initializes with an error message and error details, utilizing system information.
The error message includes the filename, line number, and error description.
Additionally, a logger records the error message in the log file.
4. Utility Functions:
save_object(file_path, obj):
Saves an object to a specified file path using dill serialization.
Handles directory creation if the specified path does not exist.
evaluate_models(X_train, y_train, X_test, y_test, models, params):Conducts model evaluation using grid search cross-validation.
Evaluates multiple models based on specified parameters.
Calculates R-squared scores for training and testing datasets.
Returns a dictionary containing the model names and corresponding test scores.

load_object(file_path):
Loads an object from a specified file path using dill deserialization.
Raises a CustomException if any error occurs during the loading process.
5. Error Handling:
Error handling is integrated into utility functions to capture and log exceptions.
CustomException instances are raised to provide detailed error messages and traceback information.
The logger records the error messages along with the timestamp in the log file.
Logging and utility functions play a crucial role in ensuring the robustness and reliability of the Python project.
Proper error handling and logging mechanisms facilitate effective debugging and troubleshooting during development and deployment stages.

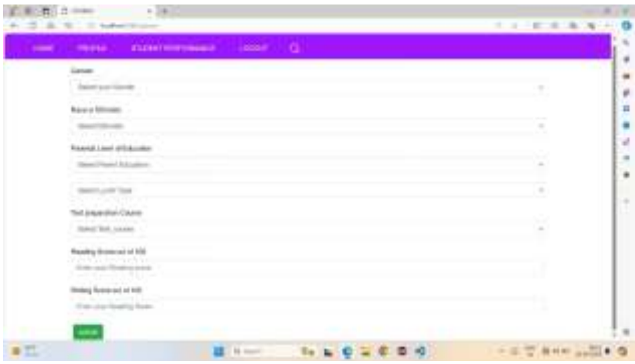
VIII.RESULTS

LOGIN PAGE

REGISTER:

MAIN PAGE

ADD CONTACT



IX. CONCLUSIONS

Student performance is a critical concern in educational contexts, presenting inherent challenges. This paper undertakes an analysis of data mining research outcomes to construct predictive models for student performance. Our study demonstrates the efficacy of machine learning algorithms in comprehending and optimizing predictive performance, particularly through data dimensionality reduction techniques like T-SNE. We incorporate four primary factors, including admission scores, first-level courses, academic achievement test (AAT), and general aptitude test (GAT), to inform our predictive models. Looking ahead, we aim to leverage deep learning architectures to further refine prediction accuracy and performance. Additionally, we intend to explore the integration of non-academic features alongside traditional academic metrics to enhance the predictive capabilities of our models.

FUTURE ENCHANCEMENT

Deep Learning Integration: Incorporate neural networks to enhance predictive modeling.
Feature Expansion: Include non-academic factors for a holistic view of student performance.
Ensemble Learning: Combine multiple models for improved prediction accuracy.
Advanced Dimensionality Reduction: Explore sophisticated techniques for data compression.
Dynamic Model Updating: Develop methods for continuous model refinement.
Personalized Prediction: Customize predictions for individual students.
Ethical Considerations: Address biases and ensure fairness in model development.
Real-Time Monitoring: Implement systems for ongoing performance tracking and intervention.

REFERENCES

[1] F. Giannakas, C. Troussas, I. Voyiatzis, and C. Sgouropoulou, ''A deep learning classification framework for early prediction of team-based academic performance,'' Appl. Soft Comput., vol. 106, Jul. 2021, Art. no. 107355.

[2] H. Hamsa, S. Indiradevi, and J. J. Kizhakkethottam, ''Student academic performance prediction model using decision tree and fuzzy genetic algo rithm,'' Proc. Technol., vol. 25, pp. 326–332, Jan. 2016.

[3] B. K. Francis and S. S. Babu, ''Predicting academic performance of students using a hybrid data mining approach,'' J. Med. Syst., vol. 43, no. 6, pp. 1–15, Jun. 2019.

[4] M. Yağcı, ''Educational data mining: Prediction of students' academic performance using machine learning algorithms,'' Smart Learn. Environ., vol. 9, no. 1, pp. 1–19, Dec. 2022.

[5] T. Le Quy, T. H. Nguyen, G. Friege, and E. Ntoutsi, ''Evaluation of group fairness measures in Student performance prediction problems,'' 2022, arXiv:2208.10625.

[6] X. Liu and L. Niu, ''A student performance predication approach based on multi-agent system and deep learning,'' in Proc. IEEE Int. Conf. Eng., Technol. Educ. (TALE), Dec. 2021, pp. 681–688.

[7] C.RomeroandS.Ventura, ''Educational data mining: A survey from 1995 to 2005,'' Expert Syst. Appl., vol. 33, pp. 135–146, Jul. 2007.

[8] R. Conijn, C. Snijders, A. Kleingeld, and U. Matzat, ''Predicting student performance from LMS data: A comparison of 17 blended courses using Moodle LMS,'' IEEE Trans. Learn. Technol., vol. 10, no. 1, pp. 17–29, Jan./Mar. 2017.

[9] S.Helal,J. Li, L. Liu, E.Ebrahimie, S.Dawson,D.J.Murray,andQ.Long, ''Predicting academic performance by considering student heterogeneity,'' Knowl.-Based Syst., vol. 161, pp. 134–146, Dec. 2018.

[10] X. Xu, J. Wang, H. Peng, and R. Wu, ''Prediction of academic perfor mance associated with internet usage behaviors using machine learning algorithms,'' J. Comput. Hum. Behav., vol. 98, pp. 166–173, Sep. 2019.