FMDB *Transactions on Sustainable Techno Learning*

Optimizing Educational Outcomes: H2O Gradient Boosting Algorithm in Student Performance Prediction

S.S. Mukhil Varmann[1,*], G. Hariprasath[2], Irina Kadirova[3]

[1,2]Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, India.
[3]Department of Training, Silk Road International University of Tourism and Cultural Heritage, Samarkand, Uzbekistan.
sm7225@srmist.edu.in[1], hg8694@srmist.edu.in[2], Uzpractice@gmail.com[3]

Abstract: Predicting students' performance is very important for educational institutions to get an insight into their academic progress, identify challenges faced by the students, and implement the targeted interventions for support and improvement. In this study, we propose using the H2O Gradient Boosting algorithm to predict student performance. The proposed algorithm offers several advantages, can handle large datasets, and is robust against overfitting. These features encompass academic records, socio-economic background, and behavioral attributes. We demonstrate the effectiveness of the H2O gradient boosting algorithm in accurately predicting student performance through experimentation and review. Our findings demonstrate considerable gains in predicting performance over usual techniques. The practical implications of our findings for educational institutions are substantial. So, Institutions can more effectively allocate resources to meet the unique requirements of students by utilizing the predictive potential of the H2O Gradient boosting algorithm to identify these individuals early on. It can increase retention rates, improve overall academic achievements, and create a friendly learning environment by taking a proactive approach to student support. H2O Grading is a boosting algorithm for improving predicting accuracy and facilitating data-driven educational decision-making.

Keywords: Student Performance Prediction; H2O Gradient Boosting Algorithm; Targeted Interventions; Socio-Economic Background; Resource Allocation; Data-Driven Decision Making; Large Dataset Handlin; Decision-Making Support.

Received on: 19/04/2023, **Revised on:** 06/07/2023, **Accepted on:** 05/09/2023, **Published on:** 22/12/2023

Cite as: S.S. Mukhil Varmann, G. Hariprasath, and I. Kadirova, "Optimizing Educational Outcomes: H2O Gradient Boosting Algorithm in Student Performance Prediction," *FMDB Transactions on Sustainable Techno Learning*., vol. 1, no. 3, pp. 165 – 178, 2023.

1. Introduction

Student overall performance analysis is the process of comparing and assessing the educational performance of college students. With the advancement of the era, gadget learning has emerged as a powerful device for studying and predicting scholarly performance. In this undertaking, we can discover the usage of the H2O prediction version for student performance analysis.H2O is an open-source system mastering platform that offers various algorithms for records analysis and predictive modeling [13]. It is specifically designed for large data and provides rapid and correct predictions. Using the H2O prediction model, we will analyze factors affecting scholar performance, including demographics, educational history, socio-economic fame, and study behavior. Training the version on historical records can examine styles and relationships between these elements and student performance [14]. One of the main benefits of using H2O for scholars' overall performance analysis is its capability to handle large datasets. As educational institutions acquire many facts about college students, H2O can efficiently

[*]Corresponding author.

system and analyze these facts to provide valuable insights. This lets educators become aware of at-hazard students and provide suitable interventions to enhance their instructional performance [15].

Another advantage of using H2O for student performance evaluation is its consumer-pleasant interface. It lets educators and researchers without a strong history in facts technology easily construct and install prediction fashions. In the end, using the H2O prediction version for student performance analysis can offer valuable insights and predictions to enhance the overall instructional performance of students [16]. Its capability to deal with massive amounts of information and use advanced system learning strategies makes it an effective tool for academic establishments. With the continuous improvement of technology, we will assume similar advancements in using H2O and different systems, gaining knowledge of gear for scholars' overall performance evaluation. Student performance analysis is a technique of comparing and gaining knowledge of the academic performance of college students. It includes reading various factors along with grades, attendance, behavior, and other applicable facts to gain insights into student fulfillment [17]. With the development of technology, gadget studying strategies have increasingly been applied in this place to improve the accuracy and performance of student performance analysis.

One popular system knowledge tool used for scholar overall performance analysis is the H2O prediction model. H2O is an open-source, disbursed system getting-to-know platform that allows immediate and correct facts analysis and modeling. It is built on top of the popular programming languages R and Python, making it without problems handy for users with one-of-a-kind programming backgrounds [18]. Using the H2O prediction model, student overall performance analysis may be achieved by utilizing huge datasets of student statistics, including grades, attendance records, and behavioral information. The model uses various algorithms to research the statistics and become aware of patterns, traits, and relationships that may be used to predict scholars' overall performance [19].

One of the key blessings of using the H2O prediction version for student overall performance analysis is its capacity to address huge datasets and complicated relationships amongst variables. This permits a more complete and accurate analysis of scholars' overall performance, leading to higher insights and choice-making for educators. Furthermore, the H2O prediction model can also make at-risk students aware of the need for additional guidance and interventions. By reading various factors, the version can predict the probability of a student suffering academically and alert educators to take necessary actions to help the scholar improve their overall performance. In conclusion, using device studying techniques, especially the H2O prediction model, in scholar performance analysis has the potential to significantly beautify our understanding of student achievement [20]. It can offer treasured insights for educators to make informed selections and interventions to guide student fulfillment. As the era develops, we can assume similar developments within systems, gaining knowledge of student performance analysis.

Student performance analysis is an important component of schooling because it enables educators and policymakers to understand the effectiveness of coaching techniques, curriculum, and other factors that affect student success. Traditionally, scholar performance evaluation has been accomplished manually through instructors and directors, which is time-consuming and vulnerable to human mistakes. However, with technological improvements, there was a growing hobby of using systems and learning strategies to analyze student performance. Machine learning is a subset of synthetic intelligence that permits structures to study from data, become aware of styles, and make selections without express programming. One of the popular gadgets that can be used to learn strategies for student performance evaluation is the H2O prediction model. H2O is an open-supply, distributed gadget gaining knowledge of a platform that permits statistics scientists and analysts to construct and install predictive fashions.

The H2O prediction version is constructed on the pinnacle of the H2O.Ai platform, which uses distributed computing to deal with big datasets and provide quicker effects. It uses numerous gadget learning algorithms, which include linear regression, decision timber, and random forests, to research data and make predictions. The H2O platform also has a user-friendly interface, making it available to fact scientists and non-technical users. The H2O prediction version follows a supervised mastering method, using a categorized dataset to teach the model and make predictions on new data. Regarding student overall performance analysis, the classified dataset could include scholar facts, such as demographic facts, grades, attendance, and other applicable statistics. The H2O version then uses these statistics to learn the styles and relationships between unique variables and make predictions about scholar performance.

Using H2O for student performance analysis offers several blessings over traditional techniques. Firstly, it allows for a more comprehensive and correct analysis of student overall performance by considering a couple of variables and their relationships. Secondly, it could identify styles and tendencies that might not be obvious via traditional techniques, providing deeper information on student performance. Thirdly, it could help become aware of at-chance college students and provide centered interventions to improve their outcomes. Finally, H2O is a consumer-friendly platform that does not require full-size coding knowledge, making it handy to a much wider range of users.

Ultimately, using the H2O prediction model in student performance analysis has numerous blessings, along with efficient and accurate analysis, figuring out at-chance students, personalized getting to know, useful resource allocation, and actual-time monitoring. However, organizations should additionally be aware of their boundaries, which include dependence on records

and amounts, restricted variables within the dataset, lack of human interplay, and the need for technical information. It is vital to be aware that the H2O prediction model ought to no longer be used as the sole method for student performance analysis but instead as a tool to complement traditional methods received.

2. Objective

- The H20 prediction version is an innovative machine getting-to-know tool that produces statistics-driven predictions about pupil overall performance. It uses state-of-the-art algorithms to investigate numerous scholarly facts, such as attendance facts, check outcomes, and extracurricular activities. This enables the version to hit upon patterns and relationships that impact scholar success.
- The H20 version leverages the strength of the device by allowing it to learn how to method large quantities of student information. It is engineered to pinpoint correlations, tendencies, and styles that can be neglected using traditional analytical techniques. The H20 version can accurately forecast students' academic performance by employing superior statistical techniques.
- A principal benefit of the H20 model is its potential to manipulate problematic, excessive-dimensional facts. It can examine numerous statistics, such as specific, numerical, and textual statistics. This permits educators to increase a nicely rounded hold close to pupil performance. Additionally, the model can manage missing information and anomalies, ensuring the evaluation is robust and reliable.
- The H20 version creates correct value determinations of student overall performance by considering several elements concurrently. It considers attendance and checks consequences, extracurriculars, and demographics to construct a holistic profile of each student's instructional path. Evaluating performance based on various factors offers educators deeper insight into pupil strengths and weaknesses.
- A key gain of the H20 model is its potential to identify underperforming college students early on. The version can flag at-hazard college students who may fall behind academically by reading ancient facts and detecting styles. This early detection permits educators to intrude promptly and deliver focused support to suffering college students.
- The H20 model's insights can tell customized interventions for college kids lacking instructional capability. Understanding the unique elements hindering a pupil lets educators craft tailored interventions addressing their precise desires. This focused approach boosts intervention effectiveness and improves academic development.

3. Review of Literature

Chen et al. [1] expressed a student performance prediction approach that considers students' common and individual characteristics. It addresses the challenge of effectively classifying student samples in multi-dimensional discrete data by combining the RMBN approach with Louvain clustering. The proposed method includes a multi-objective assessment approach and the RMHNN model to accurately predict student performance. The study emphasizes the importance of personalized student support based on predicted academic performance. The study underscores the importance of leveraging educational data mining techniques to enhance university student support and academic outcomes.

Sun et al. [2] proposed that the prediction Model and Experiment utilize Multi-Feature Fusion and an Attention Mechanism to predict student performance accurately. The model shows improved predictive ability by integrating multiple feature extraction modules, including ST-CRS, CRS-CRS, and ST-ST. The study explores the impact of data balance on model accuracy and identifies areas for improvement in predicting student grades. The research addresses challenges in collecting personal information for prediction models and emphasizes the importance of real-time factors for accurate predictions. Overall, the proposed model enhances predictive performance by considering various dimensions of student academic data.

Bujang et al. [3] presented a multiclass prediction model for student grade prediction using machine learning algorithms. The study compares the performance of various algorithms such as J48, kNN, NB, SVM, LR, and RF on real datasets. Techniques like oversampling SMOTE and feature selection address imbalanced class instances. The proposed model outperforms using oversampling SMOTE and feature selection alone, showing improved accuracy in predicting student grades. The findings address imbalanced multi-classification issues in educational settings, enhancing predictive analytics for academic performance monitoring.

Alshanqiti and Namoun [4] combine collaborative filtering, fuzzy rules, and Lasso linear regression to predict student performance. It introduces a weighted sum model to adjust the importance of the technique dynamically. Using a weighted mean scheme, the study optimizes the Self Organizing Map as a multi-label classifier. Statistical analysis shows the proposed models outperform competitors significantly. The logical rule-based model focuses on predicting student grades by analyzing past courses individually, contributing valuable insights for program leaders in higher education.

Alhazmi and Sheneamer [5] explore early prediction of students' performance in higher education using data mining and

machine learning techniques. Various models, including Random Forest and Neural Networks, classify final exam grades and predict academic success. Factors such as admission criteria, course grades, and student demographics are analyzed to enhance performance prediction. The study emphasizes the importance of using technology to mitigate risks of student failures and improve educational outcomes. Overall, the research aims to leverage data mining to better understand and predict students' academic performance.

Pelima et al. [6] delve into predicting university student graduation using machine learning algorithms. It highlights the benefits of such predictions for both students and institutions. The study emphasizes the importance of analyzing academic performance data to forecast future outcomes accurately. However, limitations such as language restrictions and publication timelines are acknowledged. Future work suggests enhancing predictive models with advanced methodologies for improved accuracy and generalization.

Butt et al. [7] explore the use of a multi-model ensemble approach to predict student performance in higher education. It aims to improve accuracy in identifying at-risk students and understanding assessment variations across different campuses. The study also investigates the impact of teacher job status on student academic success. The research provides valuable insights for enhancing student learning outcomes and decision-making in higher education.

Liu et al. [8] said that a Multiple Features Fusion Attention Mechanism enhances student performance prediction in online learning. The model improves accuracy in predicting student outcomes by integrating multiple features and utilizing an attention mechanism. The research demonstrates that the framework outperforms existing methods like DKT and IRF-DKT. The findings highlight the ability to analyze student knowledge weaknesses and create individual learning schemes. The study provides valuable insights for improving online education through predictive modeling and personalized feedback.

Alamri and Alharbi [9] focused on explainable student performance prediction models. They identified 15 articles that met their criteria out of 47 assessed. The study extracted data based on dimensions like education level and performance level. The review highlights the importance of explainable machine-learning models in predicting student performance accurately. The findings provide valuable insights for improving educational outcomes through transparent prediction models.

Jiang and Wang [10] discuss the importance of modeling knowledge states in online education for predicting student performance. It introduces a preference cognitive diagnosis approach based on students' preferred learning materials. The study compares different baselines and experimental results to evaluate the effectiveness of the proposed method. By combining the DINA model and preference degree, the method accurately predicts students' exercise scores. The research highlights the potential for improving personalized learning experiences by better understanding students' knowledge states.

Deo et al. [11] explore using modern artificial intelligence models to predict undergraduate student performance in engineering mathematics courses. The study evaluates the effectiveness of Extreme Learning Machines (ELM) in forecasting student scores across different grade categories. The authors highlight the flexibility and reliability of AI models for educational applications, emphasizing their potential for online and distance learning. The findings suggest that AI can enhance education decision-making and improve student outcomes. Overall, the study underscores the growing role of AI in higher education and its impact on student performance modeling.

Pek et al. [12] discuss using machine learning to identify at-risk students and prevent academic failure. It emphasizes the importance of early detection and intervention to support student success. The study considers factors like homework grades and students' goals for higher education in predicting academic performance. Limitations include the need for data from diverse educational settings and consideration of cultural differences. The findings suggest that machine learning can improve student outcomes and reduce failure rates.

4. Proposed methodology

4.1. H20 estimator model

H20 is an open-source platform for building machine-learning models. It offers various algorithms and tools for data pre-processing, model building, and deployment. The word estimator model In H20 refers to an algorithm that predicts outcomes based on the input data given to the model. This h2o estimator model learns patterns from historical data and uses them to make predictions on unseen data. This model is mainly used in prediction, clustering, regression, and classification. This model offers many algorithms for prediction, such as gradient boosting machines, random forests, deep learning, and many more algorithms with Python programming for conducting machine learning research.

The first in utilizing the H2O model is initializing the h20 cluster, which is done by calling a simple init() function. This init() function call initializes a local or remote H2O cluster and provides the model with the necessary computational resources

needed for the model to train the input dataset. Once the cluster has been initialized, data is imported using the import_file() function, which loads data from the file and is converted into an H2O frame. This H2O frame is very similar to Pandas Data Frame but designed for efficiently distributed data processing within the H2O platform.

H20 distributes the dataset across the nodes in the model cluster and divides the given data into smaller pieces. Each piece of data is stored and processed independently on the nodes, which enables parallel processing. This model stores data in column format, storing each column separately. This kind of storage allows access to data efficiently and can manipulate the data easily for operations like filtering, aggregation, and model training. H2O uses

Lazy evaluation, which delays the execution and computations until they are explicitly requested. This is done to optimize memory usage and avoid unnecessary computations. H2O automatically parallelizes the computations across the different nodes in the cluster. This distributed type of processing allows the H2O estimator model to handle large datasets and scale to large computational resources.

After training the model, the H2O estimator generates a performance report such as mean square Error (MSE), Root mean square error (RMSE), Mean absolute error (MAE), and Root mean squares logarithmic error (RMSLE). These evaluation metrics describe how close the model is to predicting the value. The model used by the H2O estimator in our project is a boosting machine for predicting student performance.

4.2. Gradient Boosting Machine

Gradient boosting machine is a machine learning algorithm that builds predictive models, an ensemble method combining multiple weak learners to create a strong learner. In the gradient boosting machine, boosting is achieved by fitting decision trees to the residual's other previous models. This technique is mainly used to minimize the loss function. At every iteration of the GBM algorithm, the gradient of the loss function to the predicted value is compared. Then, this gradient is used to update the predictions in the direction that minimizes the losses effectively, descending the gradient to reach the optimal solution. This algorithm's most common loss function is mean squared error (MSE) and mean absolute error (MAE). This algorithm also involves learning and regularization parameters, which control the model's learning rate and complexity. GBM uses an iterative training process, with each iteration adding a weak learner to the ensemble. In each iteration, the model is trained to minimize the residual error from the previous iteration, gradually increasing the model's performance.

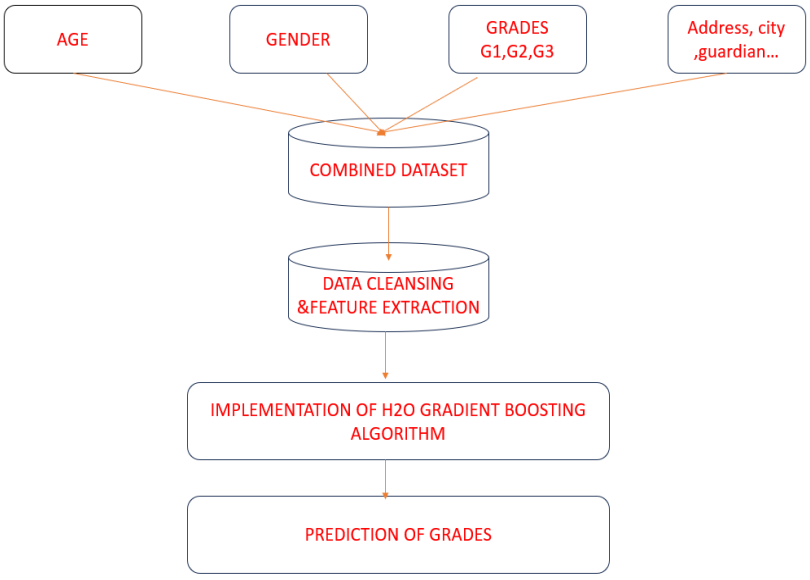


Figure 1: Architecture diagram

4.3. Random Forest classifier

A random forest classifier is a machine-learning algorithm that is used for prediction. This algorithm can handle complex data structures, capture non-linear relationships, and provide outputs. Random forest builds a collection of decision trees during the

model training phase. Each decision tree is trained on the random subset of the training data and a random subset of features. These decision trees are trained independently, and the decision trees are grown until they meet certain conditions. Random forest introduces randomness in the training process by bootstrapping the train data and randomly selecting features for each tree. This randomness helps reduce overfitting and improves the model's generalization performance. During the time of prediction, each decision tree in the forest independently predicts the class label. Random forest can provide insights into feature importance by evaluating the impact of each feature on the model's prediction. Random forest is mainly known for its scalability and ability to handle large datasets with high dimensionality. It is robust to noise and outliers in the data and performs well without extensive hyperparameter tuning. This algorithm is widely used because it can achieve high accuracy and performance generalization across a wide range of tasks and domains. It is also used for complex datasets with non-linear relationships.

4.4. Architecture diagram

Figure 1 depicts the sequence of activities and steps in the code for student performance prediction using the H2O gradient boosting algorithm. Each block in the diagram indicates a crucial step involved in the coding part. Initially, the algorithm loads and analyses the data to learn the patterns involved in the dataset, and it also learns about the parameters involved in the dataset. After the dataset is loaded into the model, it is checked for data for null values, and if any null values are present, data is filled with values using mean, median, and mode techniques. After data cleansing is performed, then data is visualized using different graph plots to understand

About the data and using different visualization techniques to gain insights into the dataset. Visualization is an important step in understanding the distribution of variables, identifying the patterns, and exploring potential relationships between variables. Plots like bar charts, boxplots, scatterplots, and heat maps are used to gain a deeper understanding of the dataset. After performing these steps, we then implement the H2O gradient boosting algorithm, with the dataset split into an 80% training set and a 20% testing dataset. After training, the model is evaluated using evaluation metrics like Mean Absolute error and root mean squared error, which are calculated to measure

Accuracy of predictions. Once the model is trained and evaluated, it is deployed into production to make predictions, known as testing. After deploying these steps, the predictive model's performance is continuously monitored to ensure accuracy. The algorithm aims to provide valuable insights into student performance, enabling educational institutions to make data-driven decisions and support student success effectively.

4.5. Formulas

Root mean squared error (RMSE): RMSE measures the square root of average squared differences between the predicted and actual values, measuring how they differ from the original value. RMSE is calculated using the below formula.

RMSE=$\sqrt{}$ $n1\sum i = 1/n$(y^i – y_i)^2)

Where:

- n is the number of samples
- y^i is the predicted value of the target variable for the data point
- y_i represents the forecasted value at the time point I generated by the model

Mean Squared Error (MSE): MSE is an evaluation metric that measures the accuracy and performance of the model's prediction. It also helps assess the model's effectiveness in capturing the underlying patterns in the data and makes reliable predictions. MSE is calculated by using the below formula:

$$\mathrm{MSE} = (1/n) * \Sigma(\hat{y}\_i - y\_i)^2$$

Where:

- n is the total number of observations
- y_i represents the actual value in the dataset
- $\hat{y}$_i represents the predicted value

Mean Absolute error (MAE): MAE is an evaluation metric used to evaluate the accuracy of predictions. It is the average magnitude of the errors predicted and the actual values. It is often used as a loss function during model training. MAE is Calculated using the following:

$$\text{MAE} = \frac{1}{\text{n}}\sum i = 1n \mid y\text{^}i - yi \mid$$

Where:

- n is the number of samples
- ŷ_i represents the predicted value
- y_i represents the actual value in the dataset

Root mean squared Logarithmic error (RMSLE)

It is also an evaluation metric used to evaluate the performance of the prediction models, which is specifically used when the target variable has a wide range of values or exhibits a skewed distribution. To calculate RMSLE, the formula below is used:

$$\text{RMSLE} = 1/n\sum i = (\log(y\text{^}i + 1) - \log(yi + 1))2$$

Where:

- n is the number of samples
- ŷ_i represents the predicted value
- y_i represents the actual value in the dataset
- log denotes the natural logarithm.

4.6. Algorithm

Step 1: Import Necessary Libraries
Import pandas, numpy, warnings, h2o, and relevant functions for metrics.

Step 2: Load Data
Read a CSV file containing student performance data in a Pandas Data Frame.

Step 3: Data Exploration
Display basic information about the dataset:
Display the first few rows of the data.
Determine the shape of the data.
Generate descriptive statistics.
Display data type information.
Display the last few rows of the data.
Check for missing values.

Step 4: Data Visualization
Use Seaborn and matplotlib to visualize the data:
Create scatter plots, box plots, and line plots to explore relationships between variables.

Step 5: Prepare Data for H2O Estimator
Transform the Data Frame into an H2O Frame.
Define the target variable and feature columns.

Step 6: Split Data into Training and Testing Sets
Define the training size (e.g., 80% of the data).
Split the data into training and testing sets.

Step 7: Create and Configure the H2O Estimator Model
Create an H2O estimator model with specific configuration settings.
Fit the model to the training data.

Step 8: Make Predictions with the H2O Estimator Model
Use the trained model to make predictions on the testing data.

Step 9: Evaluate Model Performance

Calculate various evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Precision.
Visualize the error metrics and predicted vs. actual values.

Step 10: Cross-Validation
Perform cross-validation to assess the model's performance over different data splits.
Visualize cross-validation metrics, e.g., MAE.

Step 11: End
End of the algorithm.

4.7. Execution

To implement the H2O estimator model for prediction, we first need to install the H2O library before we train our models using the command.

Pip install H2O: Then import the necessary libraries for data manipulation and visualization. Then, we will initialize the H2O estimator model so that we can train the model.

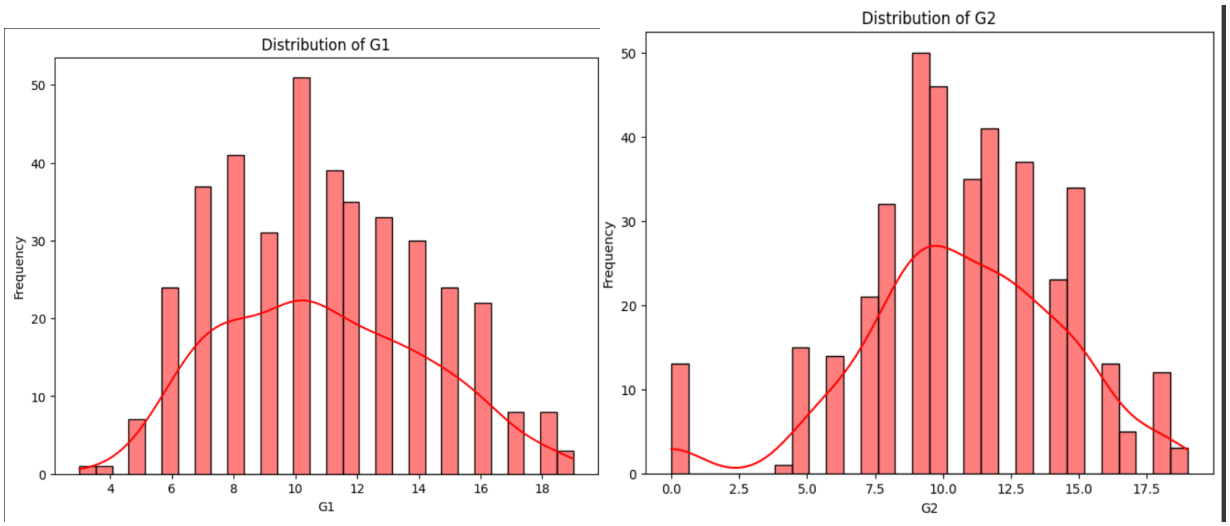
H2O.init(): Split the dataset for training and testing, 80% for training and 20% for testing. After the data is tested, we evaluate the model using evaluation metrics to decide how accurate the prediction made by the model is.

5. Implementation

5.1. Data and pre-processing

The dataset used in the student performance prediction project consists of parameters like name, gender, age, and subject grades named G1, G2, G3, and address. The dataset consists of 400 students' information. This data is given as input and checked whether any null values are present in any of the columns; after handling the null values, the data given to train the model are visualized.

5.2. Data visualization



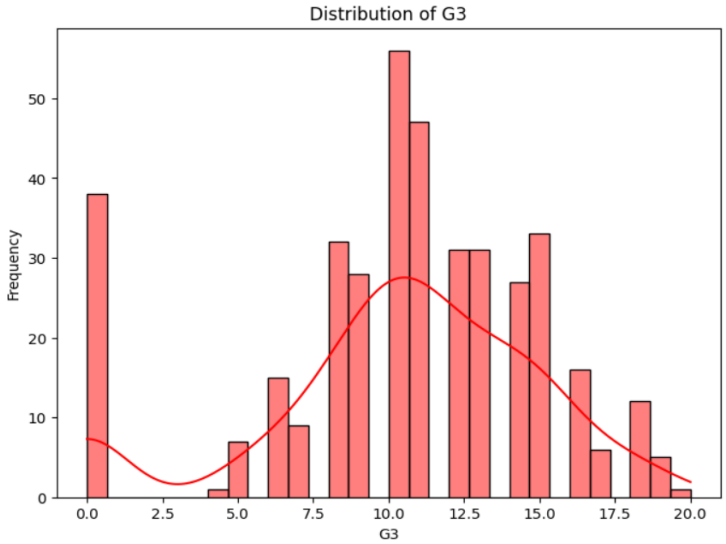


Figure 2: Distribution of different grades in the dataset

Figure 2 represents the distribution of the feature Grade (G1, G2, G3) to understand how the data is spread and the characteristics of this feature within the dataset. This histogram plot represents how the values are distributed across the ranges. This histogram visualization helps assess and select the appropriate modeling techniques and interpret the results accurately. This helps gain insights into the characteristics of the data and make informed decisions throughout the modeling process.

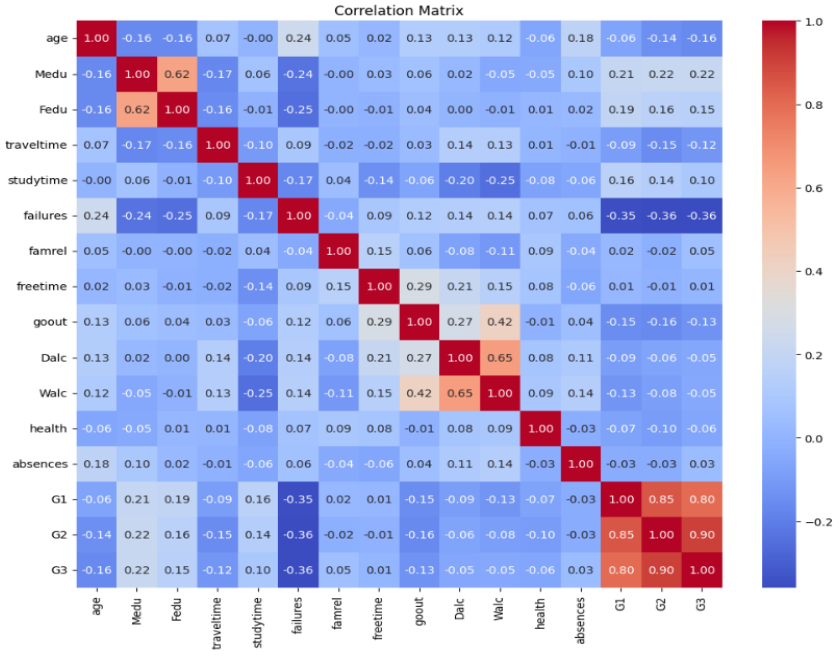


Figure 3: Correlation matrix

Figure 3 represents the correlation matrix, which helps understand the relationship between variables in a dataset. The correlation matrix is always a square matrix in which the diagonal elements represent the correlation of each variable with itself. The correlation coefficient value ranges from -1 to 1. A correlation coefficient close to 1 indicates a strong relationship, -1 indicates a string negative linear relationship, and a value close to 0 indicates there is no linear relationship. This correlation matrix provides valuable insights into the relationship between variables in a dataset, which enables better decision-making in data analysis and modeling tasks.

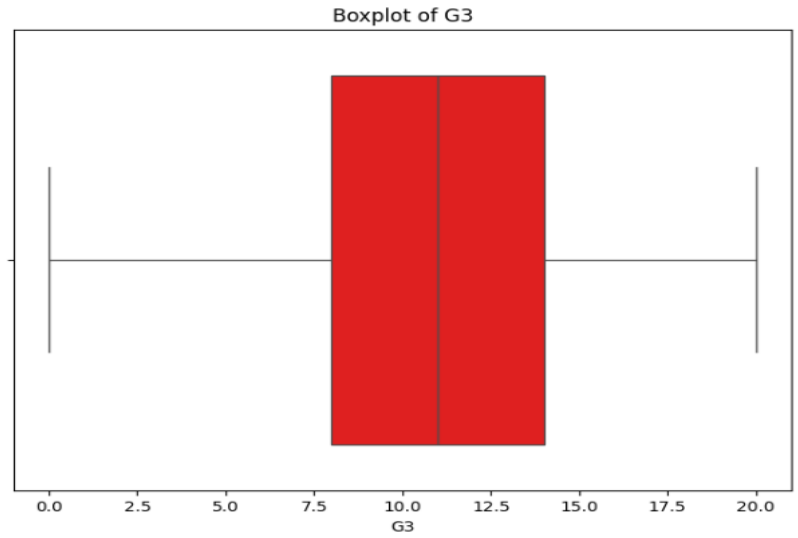


Figure 4: Box plot

Figure 4, box plot, is a graphical representation of the distribution of the dataset. The box plot is divided into three quartiles. The first quartile ranges below 25% of the data and is the median. The second quartile represents the 50th percentile of the data. The third quartile represents the 75th percentile of the data, and it is the median or upper half of the dataset. This box plot helps find out where the dataset values lie and is also used to check the outlier values.

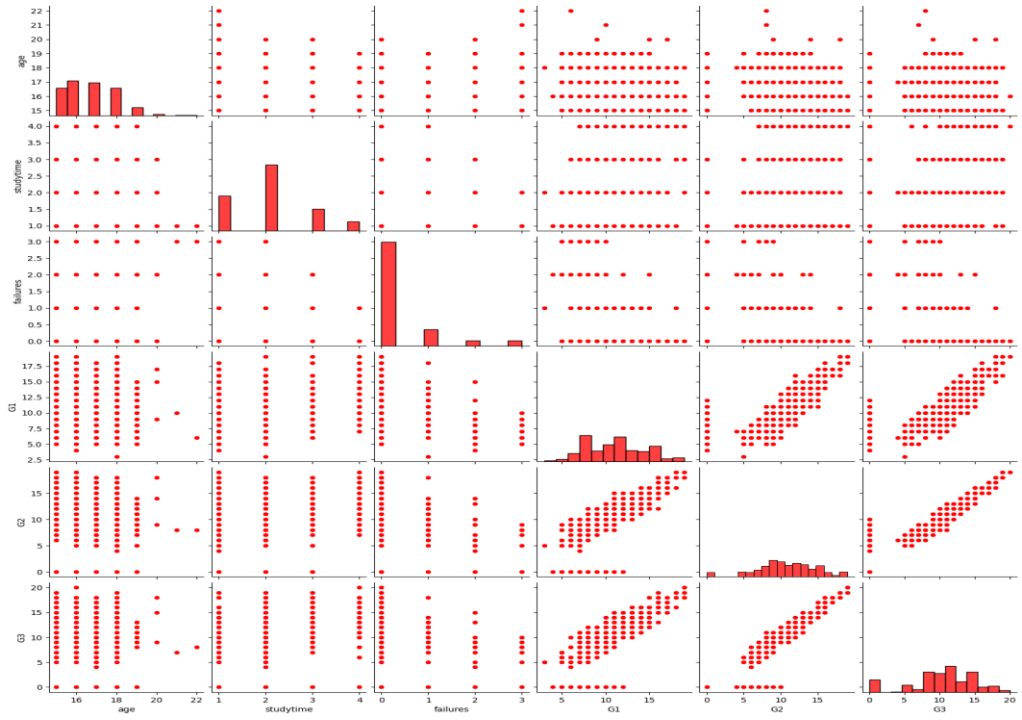


Figure 5: Pair plot

A pair plot creates a grid of scatter plots for each pair of variables in the dataset. This plot helps visualize the relationship between multiple variables. It is useful for exploratory data analysis and identifying the patterns and correlations in the data given for the model to train. This plot provides an overview of the relationship within the dataset, which helps further data analysis (Figure 5).

5.3. Training and testing

The dataset was split into training and testing data, with 80% and 20% each. The H2O gradient Boosting machine learning algorithm was used to train the model. For training, the model Google Colab was used. Python modules such as pandas, NumPy, matplotlib, and seaborn are used in experimental analysis. Google Colab provides an interactive environment where code can be written simultaneously and code can be executed in one cell. The model is trained on training data and tested on the testing data. The training and testing are done using train_test_split, imported from the library sklearn.model_selection.

5.4. Evaluation

Test data was used to test the trained models. The model performance was assessed using measures like Mean Squared Error (MSE), Root mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Residual Deviance. This evaluation section of the code allows us to evaluate the H2O model performance by evaluating the differences between projected and actual values using the abovementioned measures. The MSE and RMSE values for the student performance prediction project are 0.32 and 0.57, respectively.

6. Result and Discussion

The proposed machine learning model, developed using Python and leveraging the H2O Gradient Boosting algorithm, was evaluated rigorously to assess its predictive performance. The experiment was conducted on a system running Windows 11 with an Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz, 8GB RAM, and GTX 1650 GPU. The model's code was implemented and reevaluated using Jupyter Notebook, ensuring consistency and reproducibility.

The dataset was divided into 80% and 20% for training and testing. The model's performance was evaluated on metrics like mean absolute error, root mean squared error, and many other metrics used to evaluate the model's effectiveness and how close the predicted value is to the original value (table 1).

Table 1: Evaluation metrics for the proposed model

Metrics	Value
Mean absolute error	0.38
Mean squared error	0.32
Root mean squared error	0.57
Mean residual deviance	0.32

In contrast, when the same dataset was evaluated using the Random Forest classifier, the results, as shown in Table 2, revealed significantly higher error metrics. The Random Forest classifier yielded a mean absolute error of 1.76, mean squared error of 9.544, root mean squared error of 3.089, and mean residual deviance of 9.54.

Table 2: Comparison of proposed model and random forest classifier

Models	Mean absolute error	Mean squared error	Root mean squared error	Mean residual deviance
H2O gradient boosting model	0.38	0.32	0.57	0.32
Random forest classifier	1.76	9.544	3.089	9.54

Table 2 clearly illustrates the contrast in performance metrics between the H2O Gradient Boosting model and the Random Forest classifier, underscoring the former's superiority in accurately predicting student performance. The H2O gradient boosting algorithm's ability to handle complex relationships within the dataset and mitigate overfitting contributes to its superior predictive power.

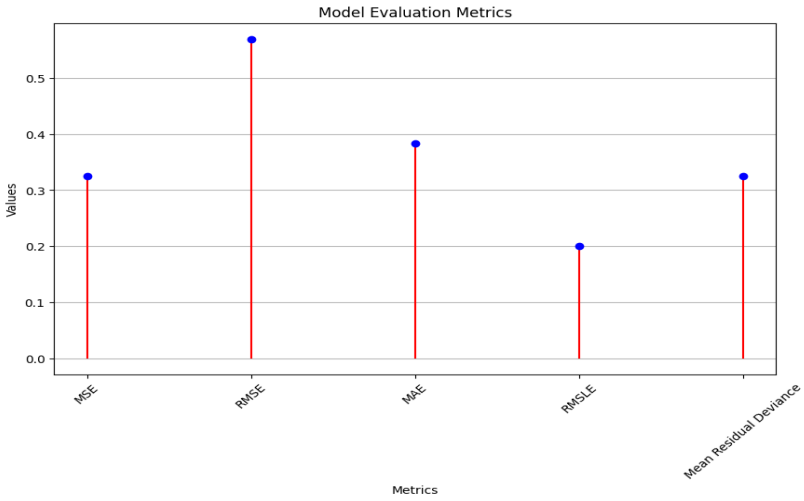


Figure 6: Graph representing the metric values

Figure 6 represents the evaluation metrics generated by the model for the test dataset. The above graph represents the mean squared error, and this project's root mean squared error is 0.32 and 0.57, respectively. The mean absolute error of the project comes out to be around 0.38, and the mean residual deviance is 0.32. These are the evaluation metrics on which project performance is evaluated, and when compared to the random forest classifier model, the evaluation metrics value of this project is so efficient.

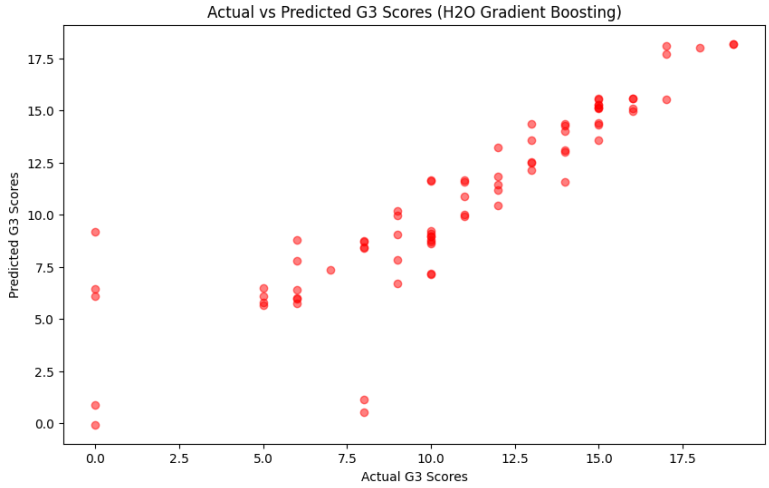


Figure 7: Graph for predicted vs actual scores

Figure 7 represents the predicted value and its difference from the original value; the dark red points indicate the predicted value, and the light red points in the graph represent the actual values. From the above graph, we can infer that the model has predicted so close to its actual value. From the above graph, we can infer that the model performs well using the H2O gradient boosting algorithm.

In conclusion, the results of this study highlight the effectiveness of the H2O Gradient boosting algorithm in predicting student performance with good evaluation metrics. By leveraging machine learning techniques, educational institutions can gain valuable insights into student progress and implement targeted interventions to support their academic performance.

7. Conclusions

The H20 prediction model provides insightful information about student performance using advanced machine learning algorithms. The model finds patterns and connections using big data sets and sophisticated algorithms. In order to create a predictive model that illustrates how many factors affect student results, several variables that have an impact on student

performance must be taken into account. Due to its intuitive design, instructors with little technical expertise may use the H20 prediction model. The model has been evaluated on various metrics and has attained a value of 0.38 as mean absolute error, 0.32 as mean squared error, and 0.57 as root mean squared error. Educators may input data, analyze findings, and acquire useful insights to enhance student performance with its user-friendly design and features. There is a great deal of promise for evaluating overall student performance in education with the H20 prediction model. Teachers may identify kids who are at risk, gather important information about the variables influencing student results, and more by utilizing cutting-edge machine learning algorithms.

Acknowledgment: The support of all my co-authors is highly appreciated.

Data Availability Statement: The research contains data related to student performance prediction and associated metrics.

Funding Statement: No funding has been obtained to help prepare this manuscript and research work.

Conflicts of Interest Statement: No conflicts of interest have been declared by the author(s). Citations and references are mentioned in the information used.

Ethics and Consent Statement: The consent was obtained from the organization and individual participants during data collection, and ethical approval and participant consent were received.

References

1. Z. Chen, G. Cen, Y. Wei, and Z. Li, "Student Performance Prediction Approach Based on Educational Data Mining," in IEEE Access, vol. 11, pp. 131260-131272, 2023, doi: 10.1109/ACCESS.2023.3335985.
2. D. Sun *et al*., "A University Student Performance Prediction Model and Experiment Based on Multi-Feature Fusion and Attention Mechanism," in IEEE Access, vol. 11, pp. 112307-112319, 2023, doi: 10.1109/ACCESS.2023.3323365.
3. S. D. A. Bujang *et al*., "Multiclass Prediction Model for Student Grade Prediction Using Machine Learning," in IEEE Access, vol. 9, pp. 95608-95621, 2021, doi: 10.1109/ACCESS.2021.3093563.
4. A. Alshanqiti and A. Namoun, "Predicting Student Performance and Its Influential Factors Using Hybrid Regression and Multi-Label Classification," in IEEE Access, vol. 8, pp. 203827-203844, 2020, doi: 10.1109/ACCESS.2020.3036572.
5. E. Alhazmi and A. Sheneamer, "Early Predicting of Students Performance in Higher Education," in IEEE Access, vol. 11, pp. 27579-27589, 2023, doi: 10.1109/ACCESS.2023.3250702.
6. L. R. Pelima, Y. Sukmana and Y. Rosmansyah, "Predicting University Student Graduation Using Academic Performance and Machine Learning: A Systematic Literature Review," in IEEE Access, vol. 12, pp. 23451-23465, 2024, doi: 10.1109/ACCESS.2024.3361479.
7. N. A. Butt, Z. Mahmood, K. Shakeel, S. Alfarhood, M. Safran and I. Ashraf, "Performance Prediction of Students in Higher Education Using Multi-Model Ensemble Approach," in IEEE Access, vol. 11, pp. 136091-136108, 2023, doi: 10.1109/ACCESS.2023.3336987.
8. D. Liu, Y. Zhang, J. Zhang, Q. Li, C. Zhang, and Y. Yin, "Multiple Features Fusion Attention Mechanism Enhanced Deep Knowledge Tracing for Student Performance Prediction," in IEEE Access, vol. 8, pp. 194894-194903, 2020, doi: 10.1109/ACCESS.2020.3033200.
9. R. Alamri and B. Alharbi, "Explainable Student Performance Prediction Models: A Systematic Review," in IEEE Access, vol. 9, pp. 33132-33143, 2021, doi: 10.1109/ACCESS.2021.3061368.
10. P. Jiang and X. Wang, "Preference Cognitive Diagnosis for Student Performance Prediction," in IEEE Access, vol. 8, pp. 219775-219787, 2020, doi: 10.1109/ACCESS.2020.3042775.
11. R. C. Deo, Z. M. Yaseen, N. Al-Ansari, T. Nguyen-Huy, T. A. M. Langlands and L. Galligan, "Modern Artificial Intelligence Model Development for Undergraduate Student Performance Prediction: An Investigation on Engineering Mathematics Courses," in IEEE Access, vol. 8, pp. 136697-136724, 2020, doi: 10.1109/ACCESS.2020.3010938.
12. R. Z. Pek, S. T. Özyer, T. Elhage, T. Özyer and R. Alhajj, "The Role of Machine Learning in Identifying Students At-Risk and Minimizing Failure," in IEEE Access, vol. 11, pp. 1224-1243, 2023, doi: 10.1109/ACCESS.2022.3232984.
13. B.R., Aravind, Bhuvaneswari, B., and S. S. Rajest, "ICT-based digital technology for testing and evaluation of English language teaching," in Handbook of Research on Learning in Language Classrooms Through ICT-Based Digital Technology, IGI Global, USA, pp. 1–11, 2023.
14. F. Wang and Z. Shen, "Research of theme-based teaching's effectiveness in English language education," Educ. Rev. USA, vol. 7, no. 7, pp. 962–967, 2023.

15. J. Padmanabhan, S. S. Rajest, and J. J. Veronica, “A study on the orthography and grammatical errors of tertiary-level students,” in Handbook of Research on Learning in Language Classrooms Through ICT-Based Digital Technology, IGI Global, USA, pp. 41–53, 2023.
16. R. S. Suman, S. Moccia, K. Chinnusamy, B. Singh, and R. Regin, Eds., “Handbook of research on learning in language classrooms through ICT-based digital technology,” Advances in Educational Technologies and Instructional Design. IGI Global, USA, 10-Feb-2023.
17. Z. Shen, H. Hu, M. Zhao, M. Lai, and K. Zaib, “The dynamic interplay of phonology and semantics in media and communication: An interdisciplinary exploration,” European Journal of Applied Linguistics Studies, vol. 6, no. 2, 2023.
18. Z. Shen, M. Zhao, and M. Lai, “Analysis of Politeness Based on Naturally Occurring And Authentic Conversations,” Journal of Language and Linguistic Studies, vol. 19, no. 3, pp. 47–65, 2023.
19. Z. Shen, M. Zhao, F. Wang, Y. Xue, and Z. Shen, “Task-Based Teaching Theory in the College English Classroom During the Teaching Procedure Targeting on the Practice of Analysis,” International Journal of Early Childhood Special Education, no. 4, 2023.
20. Z. Shen, Q. Xu, M. Wang, and Y. Xue, “Construction of college English teaching effect evaluation model based on big data analysis,” in Proceedings of the 2nd International Conference on New Media Development and Modernized Education, 2022.