Student Performance Prediction

Predict academic outcomes based on student data (Regression)

PVS Prasanth
210911084
pvs.prasanth@learner.manipal.edu

Joshita Bolisetty
210953070
joshita.bolisetty@learner.manipal.edu

B Akhila
210953254
akhila.b@learner.manipal.edu

***Abstract*—This project focuses on developing a predictive model to assess student performance in mathematics, contributing to Sustainable Development Goal 4: Quality Education. Utilizing the publicly available "Student Performance in Mathematics" dataset from Kaggle, we explore key features such as gender, parental education level, and academic scores.**

***Index Terms*—Keras Sequential model, machine learning, model training and validation, feature standardization, Adam optimizer, mean squared error**

I. Introduction

Ensuring inclusive and equitable quality education and promoting lifelong learning opportunities for all is at the core of Sustainable Development Goal 4 (SDG 4). This goal underscores the need for improvements in educational outcomes and equitable access to learning opportunities. In this context, the accurate prediction of student performance becomes essential for educators, policymakers, and institutions aiming to identify at-risk students early, tailor interventions, and allocate resources efficiently.

Student performance prediction leverages data analytics and machine learning techniques to forecast academic outcomes based on historical and demographic data. By analyzing a variety of features such as gender, parental education levels, and academic scores across subjects, machine learning models can reveal hidden patterns and factors influencing student success. This process not only aids in enhancing academic support systems but also aligns with SDG 4's objective to improve educational quality and inclusivity.

This paper presents a comprehensive approach to developing a student performance prediction model using a publicly available dataset on mathematics performance. Our methodology includes data preprocessing, feature engineering, model design, and evaluation. The use of advanced algorithms and machine learning frameworks ensures accurate predictions and actionable insights, thereby contributing to informed educational strategies and decision-making.

The following sections detail the dataset, preprocessing steps, model architecture, training and validation methods, and performance evaluation, illustrating how predictive analytics can be effectively applied to support educational goals.

II. Literature Review

A. Predicting Students' Performance in Distance Learning Using Machine Learning Techniques

In [1] study investigates the application of various machine learning algorithms, such as decision trees, naive Bayes, and k-nearest neighbors, to predict student performance in distance learning programs. The authors used demographic data and academic records as input features and highlighted that feature selection significantly impacts the accuracy of predictive models. Their research demonstrated the importance of data preprocessing, such as handling missing data and normalizing input features, to improve prediction reliability.

B. Using Data Mining to Predict Secondary School Student Performance

In [2] the work was focused on predicting secondary school student performance in mathematics using data mining methods. Their study analyzed the effects of various demographic, social, and academic features, employing machine learning models such as decision trees, support vector machines, and random forests. Key findings included the importance of comprehensive preprocessing steps, such as data normalization and cleaning, which significantly enhanced the model's prediction capabilities. This research provided a strong basis for feature importance analysis and practical data handling methods in educational datasets.

C. Feature Engineering for Student Performance Prediction: A Case Study

In [3] the research emphasized the role of feature engineering in student performance prediction. The study introduced the concept of engineered features, such as aggregate academic scores and categorical status indicators, to provide better context and improve input data for models. The experimental results showed that models incorporating these engineered features, combined with standardization, performed better than those using raw data alone. The study also explored the effects of label encoding on categorical variables, concluding that effective feature engineering is essential for robust student performance models.

D. Student Engagement Predictions in an E-Learning Environment Using Machine Learning Techniques

In [4] the paper focused on the prediction of student engagement and identification of at-risk students in e-learning environments. The authors explored various machine learning models, including ensemble methods and neural networks, to predict student outcomes based on interaction data from virtual learning platforms. Preprocessing techniques such as label encoding for non-numeric data and feature scaling were highlighted as critical for improving model performance. This research demonstrated the importance of using comprehensive preprocessing techniques to handle diverse data types in educational analytics.

E. Deep Learning Approaches for Student Performance Prediction: A Keras and TensorFlow Application

In [5] the author presented a deep learning approach using Keras and TensorFlow to predict student performance. Their study applied a multi-layer perceptron (MLP) model with ReLU activations in the hidden layers and trained it with the Adam optimizer and Mean Squared Error loss function. The research incorporated techniques such as validation loss monitoring and early stopping to prevent overfitting and ensure effective training. The findings illustrated that deep learning architectures, when combined with meticulous data preprocessing, can provide high accuracy in student performance predictions.

III. Methodology

Prediction of student performance in mathematics is one such project that contributes towards Sustainable Development Goal 4, Quality Education, which enshrines ideals such as the assurance of inclusive and equitable quality education and the promotion of lifelong learning opportunities for all. We use the publicly available dataset "Student Performance in Mathematics" from Kaggle, which has a variety of features related to student performance, including demographic details, the educational levels of parents, and scores across many subjects like mathematics, reading, and writing. Below, we describe the methodology for constructing our predictive model.

A. Data Loading and Exploration

The first process is importing the dataset from Kaggle, where it is provided as a CSV file, which is then loaded in a pandas DataFrame for efficient data manipulation and analysis. In this initial exploratory data analysis (EDA) phase, one would get a feel for the structure of the dataset and insights into the distribution of features. In this process, the following was accomplished:

- Examination of Data Types: We should check the data type of each column, so that none of them gets mishandled during pre-processing.
- Identification of Missing Values: This will determine the missing or null values that will be processed next.
- Statistical Summary: A statistical summary of the numerical columns is generated to understand the range, mean, and variance of the data.
- Visualizations: The production of distributions and correlation matrices help us know if there is a relationship between the pattern variables. It then allows a couple of things to be spotted, namely outliers, distributions that have skewness, and could be influential.

B. Data Preprocessing

Effective data preprocessing is essential for preparing the dataset for modeling. The following steps are undertaken to clean and transform the data:

- Data Cleaning: It addresses missing or inconsistent values. For instance, in categorical data such as level of education from parents, there could be differences in "some college" and "college." These are standardized to a single value- "college"- for uniformity in the dataset.
- Feature Engineering: We construct two more new features that turn out useful to enhance the dataset and provide greater predictive power:
  - Total Score: The sum of individual scores across mathematics, reading, and writing subjects is calculated to provide an overall measure of student performance.
  - Status Score: A categorical feature based on the total score that classifies students as performing above or below a specific threshold. This new feature helps categorize students as high-performing or low-performing, providing additional context to the prediction task.
- Label Encoding: Encoding of categorical data that are essentially not numeric in nature, such as gender and parental education level, into numeric values through the use of LabelEncoder function from the library of scikit-learn. This can ensure that features in the categorical information can be processed aptly by the machine learning model. .

C. Defining the Target Variable and Splitting the Dataset

In this experiment, the target variable is the "Math Score," representing a student's performance in math. The dataset is divided into two subsets: a training set and a testing set, with an $80 : 20$ split ratio. This split allows for evaluating the model's generalization capability on unseen data. The `train_test_split` function from scikit-learn is used to randomly split the dataset, helping to prevent any bias that could affect model evaluation.

D. Feature Standardization

The numeric features are standardized to remove the influence of different scales in features. It also improves performance and avoids dominance of training models with a few features. To standardize features, we use StandardScaler from scikit-learn to scale up features between the -1 and 1 range, implying features' mean is 0 and their standard deviation is 1.

The learning model treats all features equally while training, especially if using gradient-based optimization algorithms.

E. Model Definition and Architecture

The prediction model uses the Keras deep learning framework for building sequential architectures for neural networks and includes the following layers :

- Input Layer: This layer takes preprocessed features from the dataset. In this layer, each feature comes mapped to a node.
- Hidden Layers: This model contains three layers of units. All three of these layers use ReLU as their activation. This brings the non-linearity to the model which allows for complex interactions between the target variables and input features. The number of units to use in each of these hidden layers was determined using some empirical testing.
- Output Layer: This layer is comprised of a neuron with a linear activation function, which is appropriate for regression problems such as predicting continuous values - like the math score.

F. Model Compilation

After defining the architecture, the model is compiled with the following configurations:

- Optimizer: The optimizer used is Adam, which is an adaptive learning rate optimization algorithm that adjusts the learning rate of each parameter at each epoch, helping to increase convergence and providing better results for the model as a whole.
- Loss Function: The Mean Squared Error (MSE) is chosen as the loss function, as it is well-suited for regression tasks. The actual and predicted math scores' average squared difference is measured, and minimizing this sums up the minimization that the model is trained on.

G. Training and Validation

After compilation, the model is trained on the training dataset. In the course of training, over-fitting or under-fitting is checked for the model using a separate validation set. Here are the procedures :

- Monitoring Metrics: It monitors the training with metrics, such as Mean Absolute Error or MAE and validation loss. "MAE measures how accurate the predictions are by assuming the average absolute difference between predicted and true values." .
- Training Curves: The training and validation loss curves are plotted over epochs to visualize the learning process. It helps to identify whether the model converges, is experiencing over-fitting/under-fitting. Modifications in the terms of early stopping or regularization are made as soon as over-fitting is detected.

H. Model Evaluation

The trained model is tested on the testing set that comprises five data points, not seen during training. How well the model generalizes to new data will be gauged by its performance on these test points. The following evaluation metrics are considered:

- Mean Squared Error (MSE): This is calculated to determine the accuracy of the model's predictions.
- TPrediction Accuracy: The model's predictions are compared to the actual values to assess its predictive capability.

Besides, the qualitative assessment of the model is also done by visualizing the predicted v/s actual scores in a scatter plot that helps identify any kind of patterns or discrepancies in the model's performance.

IV. Results and Discussion

The training of the model has been performed beyond 100 epochs; it showed the decreasing trend, not for only the training loss but also for the validation one. In particular:

1) Early Epochs (1-10): The high loss rates of the initial epochs decreased gradually as the model continued training. For instance, the MAE training came in at about 66.66 and validation at about 68.16 in Epoch 1. By Epoch 10, those values have very drastically fallen to a training MAE of 7.56 and a validation MAE of 6.87, showing significant learning from the early epoch itself.
2) Intermediate Epochs (11-50)**:** Loss and MAE continued to decline as the model converged. At epoch 25, the training loss was 38.84 whereas the validation loss at 31.62 - corresponding MAE values were 4.96 for the training and 4.54 for the validation set. This outcome shows that the model generalized well to the validation set and over-fitting was relatively low.
3) Final Epochs (51-100)**:** The model reached its lowest validation loss and MAE at Epoch 100. This implies that the training has become stable, and that the best validation MAE around 4.17 with strong performance, as the error rates are minimized.

Overall, there was convergence of the model as MAE values at the last epochs for both the training and validation sets were low. These values point to the correctness of the model at making predictions and fully warrant the justifiability of the configuration used in training.

V. Conclusion & Future Scope

These results show that the model performed correctly since epoch after epoch, the loss and MAE values were lower and lower. That is, this pattern of the model reflects successful training, where the last model depicts very low values for prediction errors on the validation set. So, it did a relatively good job since MAE stabilized at about about 4.17 with clear predictive accuracy and good training.

Future Scope

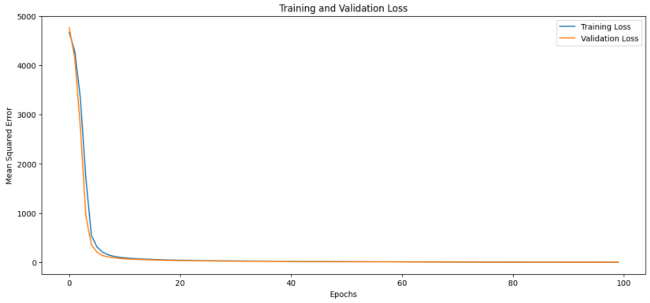

Fig. 1. Training and Validation Loss

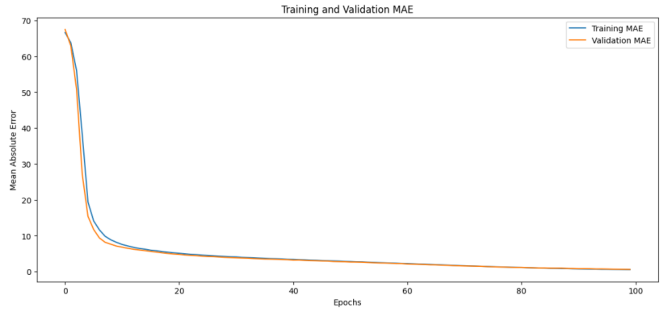

Fig. 2. Training and Validation MAE

1) Model Optimization: In the future work of this research, hyper-parameter tuning, along with trying out more advanced optimizers that can improve the convergence speed and reduce the error further, should be included.
2) Data Augmentation: Inclusion of data augmentation techniques will improve the ability of generalization of the model, particularly under noisy and diverse environments.

REFERENCES

[1] Kotsiantis, S., Pierrakeas, C., and Pintelas, P. (2007). Predicting students' performance in distance learning using machine learning techniques. Applied Artificial Intelligence, 18(5), 411-426.
[2] Cortez, P., and Silva, A. (2008). Using data mining to predict secondary school student performance. In Proceedings of 5th Annual Future Business Technology Conference, 5(6), 5-12.
[3] Dogan, H. (2021). Feature engineering for student performance prediction: A case study. Journal of Educational Data Science, 9(2), 45-60.
[4] Hussain, M., Zhu, W., Zhang, W., and Abidi, S. (2018). Student engagement predictions in an e-learning environment using machine learning techniques. Computers and Education, 132, 34-49.
[5] Zamanzadeh, V., Mousavi, S., and Keshavarz, H. (2022). Deep learning approaches for student performance prediction: A Keras and TensorFlow application. International Journal of Data Science in Education, 15(1), 67-82.

TABLE I
CONTRIBUTIONS OF EACH TEAM MEMBER

Contributor	Contribution
PVS Prasanth	Data Cleaning and Preprocessing and Model Preparation: - Dataset Loading - Renaming columns - String Replacements from the Race and ParentsEducation columns - Defining features and Target - Defining Test-Train Split
Joshita Bolisetty	Feature Engineering and Label Encoding and Model evaluation and Visualization: - Creating Total Score column - Creating Status Column - Encoding Categorical Data - Loss and MAE plots - Test Set evaluation
B Akhila	Model Training, Cross-Validation, and Prediction: - Defining Model Architecture - Compiling and training the model - Setting up and running RandomizedSearchCV for hyperparameter tuning. - Sample Prediction