Student Performance Prediction Using Hybrid Ensemble

Greeshma Chanduri, Sakshi Sheregar, Pravalika Kamineni, Aarti Deepati

University of New Haven, West Haven, CT

***Abstract*—This study uses the UCI Student Performance dataset to present a novel weighted hybrid ensemble classifier to predict academic performance. The proposed model integrates Random Forest (RF), XGBoost (XGB), and a Neural Network (NN), with weights dynamically adjusted based on the influence of prior grades (G1 and G2). Initially established at 40% RF, 30% XGB and 30% NN, the weights shift to 25% RF, 25% XGB, and 50% NN when G1 and G2 are highly influential, enhancing the model’s adaptability to grade-driven patterns. The dataset, comprising 395 students and 32 features, is pre- processed to categorize the final grades (G3) into Low (0- 9), Medium (10-14), and High (15-20). Evaluated on a 20% test split, the hybrid model achieves an accuracy of 0.82. An interactive Streamlit interface provides real-time predictions, probabilities, and feature importance, with G2 (0.3765) and G1 (0.1985) identified as dominant predictors. This work offers a practical tool for educational stakeholders and contributes a flexible ensemble approach to educational data mining.**

***Index Terms*—Machine Learning, Ensemble Classifier, Student Performance Prediction, Dynamic Weighting, Educational Data Mining**

1. Introduction and Background

Predicting student academic performance is a critical task in education, enabling early interventions to improve outcomes. Traditional methods rely on manual analysis of factors like prior grades, study habits, and socio-economic background, which is impractical for large cohorts. Machine learning offers a scalable solution by identifying patterns in historical data to forecast future performance.

The UCI Student Performance dataset [1], collected from secondary school students in Portugal, provides a rich foun- dation for such predictions. Prior studies, such as Cortez and Silva [1], have applied single models like Random Forest or Decision Trees, achieving reasonable results but lacking adaptability to varying data characteristics. Ensemble meth- ods, combining multiple classifiers, have shown promise in improving predictive power, yet most employ static weighting schemes that do not account for feature-specific influences.

This paper introduces a dynamic weighted hybrid ensemble classifier that adjusts its composition based on the influence of prior grades (G1, G2), aiming to enhance adaptability and interpretability. Integrated with a Streamlit interface, the model predicts final grades (G3) as Low (0-9), Medium (10-14), or High (15-20), offering a practical tool for educators. The study

focuses on implementing a novel ensemble methodology for educational data mining.

1. Methodology
2. *Dataset*

The UCI Student Performance dataset (Mathematics) con- tains 395 student records with 33 attributes, including demo- graphic (e.g., sex, age), socio-economic (e.g., Medu, Fedu), behavioral (e.g., studytime, absences), and academic (e.g., G1, G2, G3) features. The target variable, G3 (final grade, 0-20), is binned into three categories: Low (0-9), Medium (10-14), and High (15-20). The feature school is excluded, leaving 32 predictors.

1. *Data Preprocessing*

Data preprocessing ensures compatibility with machine learning algorithms:

* + **Categorical Encoding**: Text features (e.g., sex: “F”/“M”, Mjob: “teacher”/“other”) are converted to in- tegers using LabelEncoder.
  + **Target Binning**: G3 is discretized into [0, 1, 2] repre- senting Low, Medium, and High.
  + **Feature Selection**: 32 features are retained, excluding

school.

* + **Normalization**: Features are scaled using

StandardScaler to standardize ranges.

The dataset is split into 80% training (316 samples) and 20% testing (79 samples) sets with a fixed random seed (42) for reproducibility.

1. *Dynamic Weighted Hybrid Ensemble Classifier*

The proposed HybridEnsembleClassifier integrates three base learners:

* + **Random Forest (RF)**: 100 decision trees, each trained on random subsets of data and features, providing robust probability estimates.
  + **XGBoost (XGB)**: A gradient-boosted tree ensemble op- timizing prediction errors iteratively.
  + **Neural Network (NN)**: A multilayer perceptron with two hidden layers (100 and 50 neurons), capturing non-linear patterns.

g1\_g2\_importance = self. rf.feature\_importances\_ [g1\_idx] + self.rf. feature\_importances\_[ g2\_idx]

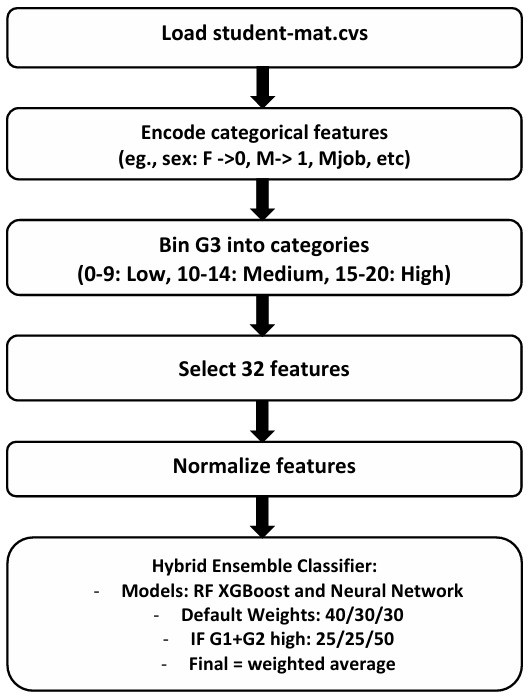
**if** g1\_g2\_importance > 0.5:

self.rf\_weight = 0.25 self.xgb\_weight =

0.25

self.nn\_weight = 0.50

**return** self

Fig. 1: Preprocessing pipeline for the UCI Student Perfor- mance dataset.

1. *Dynamic Weighting Scheme:* The hybrid model com- bines the predictions from Random Forest, XGBoost, and Neural Network by assigning each a weight that determines its influence on the final prediction. Normally, Random Forest has a stronger role (weight 40%,with XGBoost and the neural network each contributing 30%. However, if previous grades (G1 and G2) are highly influential in the data, the model increases the role of the neural network to 50% and reduces Random Forest and XGBoost to 25% each, prioritizing the ability of the neural network to model patterns related to grades.

The final prediction is computed by combining each model’s predicted likelihood for a class (Low, Medium, or High), weighted by its assigned influence, as shown in the following equation:

***P*** (class) = ***w***RF *·* ***P***RF + ***w***XGB *·* ***P***XGB + ***w***NN *·* ***P***NN (1)

where ***w***RF***, w***XGB***, w***NN are the weights for Random For- est, XGBoost, and Neural Network, respectively, and ***P***RF***, P***XGB***, P***NN are their predicted likelihoods for a given class. The class with the highest combined score is selected. This weighted combination allows the model to dynamically adapt to the characteristics of the data, balancing stability and complexity. See Listing 1 for implementation details.

Listing 1: Implementation of the dynamic weighting logic in HybridEnsembleClassifier

**def** fit(self, X, y): X\_scaled = self.scaler.

fit\_transform(X) self.rf.fit(X\_scaled, y) self.xgb.fit(X\_scaled, y) self.nn.fit(X\_scaled, y) g1\_idx = X.columns.

get\_loc(’G1’) g2\_idx = X.columns.

get\_loc(’G2’)

1. *Contributions of Individual Models and Hybrid Ratio- nale:* Each model in the hybrid ensemble contributes unique functional strengths, making the combination effective in pre- dicting student performance in diverse educational scenarios.
   * **Random Forest**: By averaging predictions from 100 decision trees trained on random subsets of data and fea- tures, Random Forest provides stability when handling di- verse student data, such as demographic, socio-economic, and behavioral factors. Its significance lies in its ability to balance multiple predictors and identify influential factors (e.g., grades, absences), offering interpretable insights for educators to understand key drivers of performance.
   * **XGBoost**: This gradient-boosted tree ensemble iteratively refines predictions by focusing on correcting errors, mak- ing it significant for capturing trends, such as consistent grade improvement or the impact of absences. Its iterative optimization ensures precise modeling of students with clear data patterns, enhancing the model’s ability to predict based on sequential academic progress.
   * **Neural Network**: With a multilayer perceptron featuring two hidden layers (100 and 50 neurons), the Neural Network excels at modeling complex, non-linear rela- tionships, particularly between prior grades (G1, G2) and final grades (G3). Its significance is its capacity to handle intricate grade-driven patterns, which are critical when prior academic performance dominates the prediction.

The hybrid model leverages these complementary strengths to create a versatile and robust prediction system. Random Forest ensures stability across varied student profiles, XG- Boost refines predictions for trend-driven cases, and the Neural Network captures complex grade relationships. This combina- tion is particularly valuable in education, where student data varies widely in structure and influence (e.g., grade-centric vs. behavioral factors). The dynamic weighting scheme, as implemented in Equation (1), allows the model to adapt its reliance on each learner based on the characteristics of the data, ensuring balanced predictions without depending on the limitations of a single model. This adaptability supports tailored interventions, such as identifying at-risk students or guiding resource allocation, making the hybrid approach ideal for educational applications.

1. *Evaluation Metrics*

Performance is assessed on the test set using:

* + **Accuracy**: Proportion of correct predictions.
  + **Precision, Recall, F1-Score**: Per-class metrics to evaluate prediction quality for Low, Medium, and High.

1. *Streamlit Interface*

A Streamlit application enables user interaction:

* + **Input**: 32 features via dropdowns, sliders, and check- boxes.
  + **Output**: Predicted grade, probabilities, and top-10 feature importance.

1. Results and Discussion
2. *Model Performance*

The hybrid ensemble was evaluated on the 79-sample test set, achieving an accuracy of 0.82. The model effectively predicts Low, Medium, and High grades, as shown in the classification metrics below, supporting its use in educational settings for identifying student performance levels.

TABLE I: Classification Metrics for Hybrid Ensemble

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | Precision | Recall | F1-Score | Support |
| Low | 0.86 | 0.89 | 0.87 | 27 |
| Medium | 0.74 | 0.88 | 0.80 | 32 |
| High | 1.00 | 0.65 | 0.79 | 20 |

The macro average F1-score of 0.82 and weighted average F1-score of 0.82 indicate balanced performance across classes, reinforcing the model’s ability to handle diverse student out- comes.

1. *Feature Importance*

The top-10 features from Random Forest in the hybrid model highlight key predictors:

TABLE II: Top-10 Feature Importance

|  |  |
| --- | --- |
| Feature | Importance |
| G2 | 0.3765 |
| G1 | 0.1985 |
| absences | 0.0404 |
| age | 0.0276 |
| failures | 0.0240 |
| health | 0.0239 |
| goout | 0.0230 |
| Mjob | 0.0195 |
| Fedu | 0.0195 |
| Medu | 0.0190 |

Prior grades G2 and G1 dominate, with a combined im- portance of 0.575 (0.3765 + 0.1985), triggering the dynamic weight adjustment to prioritize the Neural Network, as de- scribed in Section II.

1. *Discussion*

The dynamic weighting scheme enhances adaptability by prioritizing the Neural Network when grades dominate, lever- aging its strength in modeling complex grade patterns. The weighted combination in Equation (1) ensures the model balances stability, precision, and complexity, making it suitable for diverse educational data. The Streamlit interface adds

practical value, enabling real-time exploration of predictions and insights for educators.

Limitations include reliance on a single dataset and potential overfitting to G1/G2 dominance. Future work could incor- porate cross-validation or additional datasets (e.g., student- por.csv) to validate generalizability.

1. Conclusion

This study proposes a dynamic weighted hybrid ensemble classifier that adapts its weights based on the influence of prior grades, achieving an accuracy of 0.82 in predicting student performance. Key predictors G2 (0.3765) and G1 (0.1985) drive the model’s success, supported by an intuitive Streamlit interface. By integrating Random Forest, XGBoost, and Neural Network with dynamic weighting, the model offers a flexible, interpretable tool for educational data mining, with potential for broader application in performance prediction tasks.

Future enhancements could explore automated weight opti- mization and multi-dataset validation to strengthen robustness.

References

1. P. Cortez and A. Silva, “Using Data Mining to Predict Secondary School Student Performance,” in *Proceedings of 5th FUture BUsiness TEChnology Conference*, pp. 5–12, 2008.
2. UCI Machine Learning Repository, “Stu- dent Performance Dataset,” [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Student+Performance.
3. Scikit-learn: Machine Learning in Python, https://scikit-learn.org.
4. XGBoost Documentation, https://xgboost.readthedocs.io.
5. Streamlit: The fastest way to build and share data apps, https://streamlit.io.