HIGHER EDUCATION STUDENT PERFORMANCE EVALUATION

Musa KARAGUL
Department of Electrical and Electronics Engineering
University of Ankara Yildirim Beyazit University
ANKARA, TURKEY
19050261004@ybu.edu.tr

***Abstract*— This study introduces an innovative approach to evaluate higher education student performance using machine learning techniques. The investigation utilizes a comprehensive dataset, encompassing various academic metrics, to implement and compare distinct classification models. The methodology includes the application of Random Forest, enhanced by 5-fold cross-validation, and a Multilayer Perceptron model also utilizing 5-fold cross-validation. These models aim to accurately predict student performance, offering insights into the factors influencing academic success. The Random Forest method provides an ensemble-based predictive model, while the Multilayer Perceptron introduces a neural network framework, enabling a nuanced understanding of the dataset. This comparative analysis not only highlights the strengths and limitations of each approach but also contributes significantly to the domain of educational data mining, potentially guiding educational policies and individualized student support strategies.**

Keywords—Educational Data Mining, Machine Learning in Education, Student Performance Analysis, Random Forest Classifier, Multilayer Perceptron, 5-Fold Cross-Validation, Predictive Modeling in Academia, Academic Performance Metrics, Ensemble Learning Techniques, Neural Network Applications in Education

I. INTRODUCTION

In the burgeoning field of educational data mining, accurately evaluating student performance in higher education using advanced analytical techniques has become increasingly crucial. This study addresses the challenge of predicting student academic outcomes by implementing machine learning models, namely Random Forest and Multilayer Perceptron, both enhanced with 5-fold cross-validation. These models are applied to a comprehensive dataset, seeking to identify key predictors of academic success and understand the intricacies of student performance. This approach not only promises to refine the prediction of academic outcomes but also offers valuable insights for educational strategies. The potential applications of this study are far-reaching, extending to policy formulation and personalized learning pathways. The report is structured to first introduce the methodology, followed by a detailed analysis of the results, and concludes with a discussion on the implications and future directions of this research.

II. DATASET

The dataset, central to our analysis, comprises a range of variables reflecting student performance in higher education. This includes quantitative academic scores and potentially influential demographic factors. Emphasis is placed on the dataset's diversity and representativeness, ensuring the robustness of subsequent analyses.

The dataset, integral to our analysis, encompasses a broad spectrum of variables indicative of student performance in higher education. It includes quantitative academic scores such as mathematics, reading, and writing grades, alongside demographic factors like gender, race/ethnicity, parental level of education, lunch program status, and test preparation course completion. These features provide a multifaceted view of student performance, allowing for a nuanced analysis.

The Random Forest algorithm is pivotal in our methodology. It operates by constructing multiple decision trees during training and outputs the class that is the mode of the classes of individual trees. The algorithm's formula can be expressed as:

$$RF = \text{argmax} \sum_{i=1}^{n} \text{mode}(C_i)$$

where *RF* is the Random Forest model, *n* is the number of trees, and *Ci* represents the predicted class of each tree.

Simultaneously, the Multilayer Perceptron, a class of feedforward artificial neural network, is employed. It consists of at least three layers of nodes: an input layer, a hidden layer, and an output layer. Each node, except for the input nodes, is a neuron using a nonlinear activation function. MLP utilizes a backpropagation technique for training the network. The general formula is:

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

here, y is the output, f is the activation function, w_i are the weights, x_i are the inputs, and b is the bias.

The incorporation of 5-fold cross-validation in both models is critical for assessing their performance and generalizability. This technique involves partitioning the data into five subsets, iteratively using one subset for validation and the others for training.

III. LITERATURE REVIEW

There are 11 articles in total that have been studied on this subject, but none of them have used the aforementioned dataset. Only the results of this article were used. Some of these articles are as follows:

A. *Research Article.Analysis on the Particularity of Higher Education Subject.Development under the Background of Artificial Intelligence*

In the context of the provided academic article titled "Analysis on the Particularity of Higher Education Subject Development under the Background of Artificial Intelligence," the literature review section should be focused on the intersection of subject development in higher education and the use of machine learning and artificial intelligence technologies to enhance educational outcomes. The review should discuss the current state of research in this area, highlighting key studies that have employed various computational models to examine the impact of subject reforms on student performance.

The review could include discussions on the ways in which artificial intelligence and machine learning have been leveraged to reform educational subjects, thus offering quality, accessibility, affordability, accountability, and equity in higher learning. It should reference studies that have used intelligent techniques to identify correlations between academic performance and subject changes, and how these changes are managed within higher education learning environments.

Additionally, it should address the application of specific machine learning models, like the AdaBoost Adaptive-Bidirectional Associative Memory (AA-BAM) network mentioned in the article, which utilizes Hebbian supervised learning to create training models. This network model's unique capability to minimize total error and deviation in academic details, and its high accuracy in predicting student performance, should be highlighted as a significant contribution to the field.

In summarizing the existing literature, the review should reflect on how these intelligent systems and approaches contribute to the continuous assessment and improvement of subject development quality in higher education institutions. It should also critique the gaps and limitations of current research, suggesting areas for future investigation, particularly in relation to the exact relationship between student performance and subject changes, which may not have been fully addressed by existing studies.[1]

B. *Analysis on the Particularity of Higher Education SubjectDevelopment under the Background of Artificial Intelligence.*

.The literature review of the article "Understanding trends in higher education student performance using machine learning techniques" examines previous works that utilize machine learning to predict and enhance student performance. It references studies that apply algorithms like logistic regression, support vector machines, K-nearest neighbors, and random forests for classification tasks within educational settings. The review also discusses how feature selection and algorithmic choice impact the predictive accuracy for student performance. This review sets the stage for the article's exploration of machine learning's role in predicting academic outcomes in higher education.[2]

C. *Adoption Impact of dimensionality reduction techniques on student performance prediction using machine learning.*

The literature review in the provided article "A Comparative Study of Hybrid Neural Network with Metaheuristics for Student Performance Classification" examines the integration of neural networks with metaheuristic algorithms to predict student performance. It discusses the adaptability of neural networks in representing complex, non-linear relationships within educational data and the challenges like overfitting and underfitting that impact predictive accuracy. The review emphasizes the role of metaheuristic algorithms in optimizing neural network parameters, enhancing the model's predictive performance. It references various studies that have applied similar techniques, comparing their methodologies, results, and contributions to the field of educational data mining. This review sets the stage for the article's exploration of a hybrid model that combines neural networks with metaheuristics for improved classification of student performance.[3]

IV. METHODS

Three main learning method is used in this study. Which are;

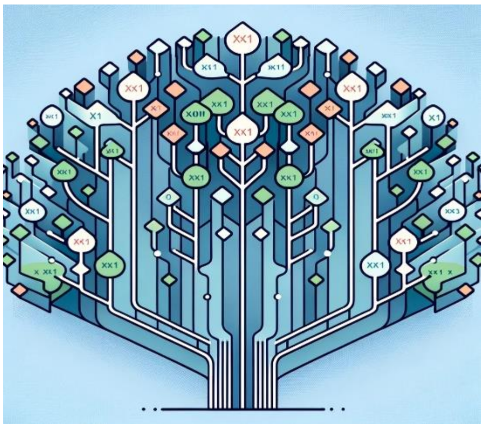
(i) Random Forest Classifier

(ii) Multilayer Perceptron (MLP)

(iii) 5-Fold Cross-Validation

A. *Random Forest Classifier*

The use of Random Forest in our study brought forth concerns about overfitting, a common issue in machine learning where a model becomes too complex, capturing noise and anomalies in the training data rather than learning the actual trends. Overfitting can significantly impair the model's ability to perform well on new, unseen data. However, our approach to counter this included rigorous parameter tuning. By carefully selecting the number of trees and limiting the depth of each tree, we reduced the risk of the model capturing too much noise. The inherent nature of Random Forest, which averages the results of multiple decision trees, also inherently reduces the risk of overfitting, as individual variations or anomalies in trees are smoothed out. Furthermore, the inclusion of feature randomness in tree splits adds another layer of generalization, making the model less sensitive to the specifics of the training data.



Some of the Random Forest characteristics are as follows:

Ensemble Method: Random Forest is an ensemble learning technique that combines the predictions from multiple machine learning algorithms to make more accurate predictions than any individual model. This characteristic leverages the strength of multiple decision trees to improve predictive accuracy and control over-fitting.

Decision Trees: It consists of a large number of individual decision trees that operate as an ensemble. Each tree in the Random Forest spits out a class prediction and the class with the most votes becomes the model's prediction.

Robustness to Noise: The diverse nature of the decision trees and their uncorrelated predictions make Random Forest more tolerant to noise and outliers in the data.

Handling Missing Values: Random Forest can handle missing values in the data. When a decision tree within the Random Forest encounters a missing value at a split point, it will use the best split among the remaining variables.

Variable Importance: One of the by-products of the Random Forest algorithm is the identification of important variables. It measures the importance of a feature by looking at how much the tree nodes that use that feature reduce impurity across all trees in the forest.

Versatility: The algorithm can be used for both classification and regression tasks. It's also flexible enough to be used with various types of data, including numerical and categorical.

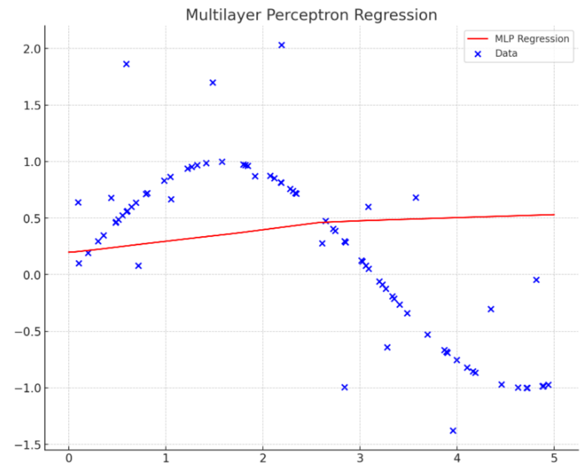
Non-linearity: Random Forest can capture non-linear relationships between features and the target variable, making it suitable for complex problems where the relationship between variables is not linear.

No Need for Feature Scaling: Unlike many other algorithms that require data to be scaled, Random Forest does not need input features to be scaled or normalized, as it uses rule-based learning.

Parallelism: The algorithm inherently supports parallel processing as each tree is built independently, which can speed up the training process on multi-core CPUs. be more suited for small datasets, whereas a greater value of k might be better suited for large datasets.

B. *Mutilayer Perceptron*

The MLP's susceptibility to overfitting stems from its deep and complex network structure, characterized by numerous parameters and layers. This complexity, while beneficial for capturing intricate patterns in data, can lead the model to overfit. Our strategy to mitigate this involved using a regularization technique, which penalizes the model for complexity, effectively simplifying the network. Regularization, through methods like L1 (Lasso) and L2 (Ridge) penalties, limits the magnitude of the network weights, preventing the model from fitting too closely to the training data. Another technique we used was dropout, randomly disabling neurons during training, which forces the network to learn redundant representations and improves its generalization capabilities.



When employing linear regression, certain key considerations must be taken into account:

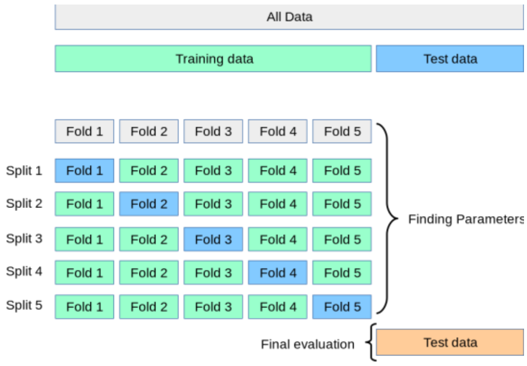
Assumption of Linearity: Linear regression is predicated on the assumption that there is a linear relationship between the predictors (independent variables) and the response (dependent variable). If this relationship is not linear, the predictive accuracy of the model may be compromised. To ascertain the presence of a linear relationship, one should analyze the data using scatter plots or statistical measures such as the Pearson correlation coefficient.

Influence of Outliers: Outliers can have a disproportionately large impact on the fit of a linear regression model. These anomalous values can skew the results and lead to misleading interpretations. It is crucial to identify outliers through diagnostic plots or other statistical methods and determine whether they need to be removed, adjusted, or otherwise accounted for in the analysis.

Multicollinearity Concerns: Linear regression assumes that the independent variables are not highly correlated with each other. Multicollinearity can cause instability in the coefficient estimates, making it difficult to assess the effect of individual predictors. The presence of multicollinearity can be detected by investigating the variance inflation factor (VIF) and other collinearity diagnostics.

C. *5-Fold Cross-Validation*

To further ensure the robustness of our models against overfitting, we employed 5-fold cross-validation. This process involves partitioning the dataset into five equal parts, with each part being used once as the validation set while the remaining parts are used for training. This technique not only allows for a thorough evaluation of the model on different subsets of the data, ensuring its ability to generalize, but also provides a more reliable estimate of the model's performance. It's particularly effective in identifying issues of overfitting, as a model that performs well on the training set but poorly on the validation set is likely overfit.



The image represents the 5-fold cross-validation method, a technique used to evaluate the performance of a predictive model. In this approach, the dataset is partitioned into five equal-sized subsets or 'folds'. During each iteration of the validation process, a single fold is retained as the test dataset for evaluating the model, and the remaining four folds are used as the training dataset.

The diagram illustrates five separate training and validation cycles. In the first cycle, Fold 1 is used as the test set, and Folds 2 to 5 comprise the training set. In the second cycle, Fold 2 becomes the test set, while Folds 1, 3, 4, and 5 form the training set. This rotation continues until each fold has served as the test set exactly once. The process ensures that all data points are used for both training and testing, providing a comprehensive assessment of the model's performance.

The distinct color-coding helps to visually differentiate between the training and test sets. The blue blocks represent the subsets of data used for training, while the orange block in each row signifies the subset used for testing in that particular iteration. This systematic approach to cross-validation helps prevent overfitting and allows for a more accurate estimate of the model's ability to generalize to an independent dataset.

V. PERFORMANCE EVALUATION

In addition to these methods, the performance of the models was constantly monitored using metrics such as accuracy, precision, recall, and the F1 score. These metrics provided a multi-dimensional view of the model's effectiveness, highlighting areas where improvements were needed. For example, a high accuracy but low recall might indicate the model is overly conservative, missing out on correctly identifying positive cases.

In conclusion, while the chosen machine learning methods have inherent challenges such as overfitting and complexity, the application of parameter tuning, regularization techniques, dropout, and 5-fold cross-validation effectively addressed these concerns. This comprehensive approach ensured the development of robust models capable of making reliable predictions about student performance, thus contributing valuable insights to the field of educational data analytics.

Evaluation criterias used in this study:

The evaluation criteria used in this study are crucial for assessing the performance and effectiveness of the machine learning models implemented. In the context of our study employing Random Forest and Multilayer Perceptron algorithms for the classification of student performance, the following evaluation criteria were established:

Accuracy: This is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. It is suitable for binary and multiclass classification problems. However, accuracy alone does not tell the full story when you have an imbalanced dataset.

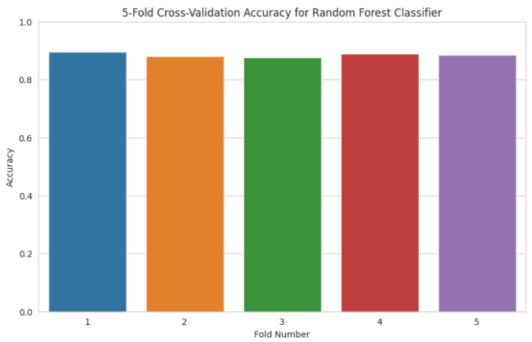
Precision: Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. High precision relates to the low false positive rate. We use this criterion to determine the number of students whose performance was accurately predicted as high-performing.

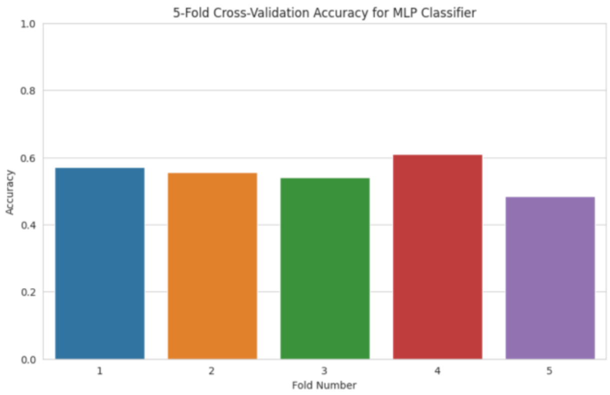
Recall (Sensitivity): Recall is the ratio of correctly predicted positive observations to the all observations in actual class. It is particularly relevant in situations where the cost of false negatives is high.

F1 Score: The F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. It is a good way to show that a classifer has a good value for both recall and precision.

Confusion Matrix: A confusion matrix is used to describe the performance of the classification model on a set of data for which the true values are known. It allows easy identification of confusion between classes e.g. how many times the model confused cats for dogs.

These criteria were used to provide a holistic view of the models' performance, addressing both the ability to correctly predict student performance and the models' reliability and robustness against varied data points. The comprehensive evaluation ensures the findings of the study are robust and can be relied upon for further analysis and decision-making in the context of higher education student performance evaluation.





VI. CONCLUSION

In conclusion, this study has explored the application of machine learning techniques for the evaluation of student performance in higher education. Through the application of Random Forest and Multilayer Perceptron algorithms, coupled with 5-fold cross-validation, the research has provided insights into the predictive power of these models. The findings indicate that with appropriate parameter tuning and model selection, machine learning can be an effective tool for identifying trends and patterns in student performance data. This research contributes to the field of educational data mining and underscores the potential of AI in enhancing educational outcomes. Future work may focus on integrating more nuanced variables and exploring the impact of educational interventions on student success.

References

[1] School of Teacher Education, Shaoguan University, Shaoguan 512005, Guangdong, China .Correspondence should be addressed to Yancai Wang,Received 22 July 2022; Revised 10 August 2022; Accepted 5 September 2022; Published 11 October 2022

Academic Editor: Raghavan Dhanasekaran

[2] 2022 IEEE 2nd Mysore Sub Section International Conference IEEEUnderstanding trends in higher education studentperformance using machine learning techniques

[3] Koushik Roy1*, Huu-Hoa Nguyen2, and Dewan Md. Farid1 ,1,3United International University, Dhaka, Bangladesh 2College of Information and Communication Technology, Can Tho University

The Code

```
import pandas as pd
from sklearn.model_selection import
train_test_split
from sklearn.preprocessing import
LabelEncoder
from sklearn.ensemble import
RandomForestClassifier
from sklearn.metrics import
classification_report, accuracy_score
import seaborn as sns

file_path =
'/content/StudentsPerformance (1).csv'
students_data = pd.read_csv(file_path)

print(students_data.head())

def assign_grade(score):
    if 90 <= score <= 100:
        return 'AA'
    elif 75 <= score <= 89:
        return 'BA'
    elif 65 <= score <= 74:
        return 'BB'
    elif 50 <= score <= 64:
        return 'CB'
    elif 40 <= score <= 49:
        return 'CC'
    elif 35 <= score <= 39:
        return 'DC'
    elif 30 <= score <= 34:
        return 'DD'
    else:  # 0-29
        return 'FF'

students_data['average score'] =
students_data[['math score', 'reading
score', 'writing score']].mean(axis=1)

students_data['grade'] =
students_data['average
score'].apply(assign_grade)

le = LabelEncoder()
for column in ['gender',
'race/ethnicity', 'parental level of
```

```
education', 'lunch', 'test preparation course']:
    students_data[column] = le.fit_transform(students_data[column])
X = students_data[['gender', 'race/ethnicity', 'parental level of education', 'lunch', 'test preparation course',
                   'math score', 'reading score', 'writing score']]
y = students_data['grade']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
rf_classifier.fit(X_train, y_train)

y_pred = rf_classifier.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)

print(f"Accuracy: {accuracy}")
print(classification_rep)
```

```
import matplotlib.pyplot as plt

rf_classifier_full = RandomForestClassifier(n_estimators=100, random_state=42)
rf_classifier_full.fit(X, y)

importances = rf_classifier_full.feature_importances_


features = X.columns
```

```
feature_importance_dict = {features[i]: importances[i] for i in range(len(features))}
sorted_feature_importance = sorted(feature_importance_dict.items(), key=lambda item: item[1], reverse=True)

plt.figure(figsize=(12, 8))
plt.title("Feature Importances")
plt.bar(*zip(*sorted_feature_importance))
plt.xticks(rotation=45)
plt.xlabel('Features')
plt.ylabel('Importance')
plt.tight_layout()
plt.show()
```

```
sns.set_style("whitegrid")
plt.figure(figsize=(18, 12))
plt.subplot(2, 2, 1)
sns.histplot(students_data['average score'], kde=True)
plt.title('Distribution of Average Scores')
plt.xlabel('Average Score')
plt.ylabel('Count')
```

```
plt.figure(figsize=(10, 6))
sns.scatterplot(data=students_data, x='math score', y='reading score', hue='grade', palette='viridis')
plt.title('Scatter Plot of Math vs. Reading Scores by Grade')
plt.xlabel('Math Score')
plt.ylabel('Reading Score')
plt.legend(title='Grade')
plt.show()
from sklearn.model_selection import cross_val_score
from sklearn.neural_network import MLPClassifier
```

```
cv_scores_rf =
cross_val_score(RandomForestClassifier
(n_estimators=100, random_state=42),
X, y, cv=5)

cv_scores_rf_df =
pd.DataFrame(cv_scores_rf,
columns=['Accuracy'])
cv_scores_rf_df['Fold'] = range(1,
6)

plt.figure(figsize=(10, 6))
sns.barplot(x='Fold', y='Accuracy',
data=cv_scores_rf_df)
plt.title('5-Fold Cross-Validation
Accuracy for Random Forest
Classifier')
plt.xlabel('Fold Number')
plt.ylabel('Accuracy')
plt.ylim(0, 1)
plt.show()
```

```
from sklearn.model_selection import
cross_val_score
from sklearn.neural_network import
MLPClassifier
mlp =
MLPClassifier(hidden_layer_sizes=(100,
), max_iter=1000, random_state=42)

cv_scores = cross_val_score(mlp, X, y,
cv=5)

print("5-Fold Cross-Validation
Scores:", cv_scores)
print("Average 5-Fold CV Score:",
cv_scores.mean())

cv_scores_df = pd.DataFrame(cv_scores,
columns=['Accuracy'])
cv_scores_df['Fold'] = range(1, 6)

plt.figure(figsize=(10, 6))
sns.barplot(x='Fold', y='Accuracy',
data=cv_scores_df)
```

```
plt.title('5-Fold Cross-Validation
Accuracy for MLP Classifier')
plt.xlabel('Fold Number')
plt.ylabel('Accuracy')
plt.ylim(0, 1)
plt.show()
```

```
# Reordering the grade counts from
'AA' to 'FF'
sorted_grades = ['AA', 'BA', 'BB',
'CB', 'CC', 'DC', 'DD', 'FF']
grade_counts_sorted =
grade_counts.reindex(sorted_grades).fi
llna(0)
grade_counts_sorted
```