**JTAM (Jurnal Teori dan Aplikasi Matematika)**
http://journal.ummat.ac.id/index.php/jtam
p-ISSN 2597-7512 | e-ISSN 2614-1175
Vol. 8, No. 3, July 2024, pp. 653-666

# Effectiveness of Machine Learning Models with Bayesian Optimization-Based Method to Identify Important Variables that Affect GPA

**Arifuddin R[1], Utami Dyah Syafitri[1], Erfiani[1]**
[1]Departement of Statistics, IPB University, Bogor, Indonesia
arif1996arifuddin@apps.ipb.ac.id

**Article History:**
Received : 24-01-2024
Revised : 29-04-2024
Accepted : 29-04-2024
Online : 17-07-2024

**Keywords:**
GPA;
Bayesian Optimization;
Random Forest;
XGBoost.

**ABSTRACT**

To produce superior human resources, the SPs-IPB Master Program must consider the factors influencing the GPA in the student selection process. The method that can be used to identify these factors is a machine learning algorithm. This paper applies the random forest and XGBoost algorithms to identify significant variables that affect GPA. In the evaluation process, the default model will be compared with the model resulting from Bayesian and random search optimization. Bayesian optimization is a method for optimizing hyperparameters that combines information from previous iterations to improve estimates. It is highly efficient in terms of computing time. Based on a balanced accuracy and sensitivity metrics average, Bayesian optimization produces a model superior to the default model and more time-efficient than random search optimization. XGBoost sensitivity metric is 25% better than random forest. However, random forest is 19% better in accuracy and 30% in specificity. Important variables are obtained from the information gain value when splitting the tree nodes formed. According to the best random forest and XGBoost model, variables that have the most influence on students' GPA are Undergraduate University Status (X8) and Undergraduate University (X6). Meanwhile, the variables with the smallest influence are Gender (X4) and Enrollment (X9).

https://doi.org/10.31764/jtam.v8i3.21711



## A. INTRODUCTION

One of the goals of higher education, according to Law NO. 12 of 2012, is to produce graduates who master science and technology so that they can contribute to national interests and increase the nation's competitiveness. Therefore, higher education plays a vital role in creating the character of an advanced, inclusive, and just society through superior human resources (Žalėnienė & Pereira, 2021). In producing human resources of superior character, the IPB master's program needs to carry out strict selection to filter out the best student candidates. A person's academic achievement generally reflects the quality of human resources. At the higher education level, academic achievement is called Grade Point Average (GPA) (Caraka & Sugiarto, 2017).Two factors influence academic achievement, namely internal and external factors (Beltrán-Velasco et al., 2021). Internal factors include psychological, cognitive, motivational, and learning methods, while external factors include social and economic conditions and educational history. Many studies show a large influence of external factors, such as socioeconomic and demographic disparities, on learning outcomes (Cheng & Kaplowitz, 2016).

Machine learning methods have been applied in various fields to analyze large amounts of data, one of which has been developed in the education sector (Fahd et al., 2022). In Ahmed and Khan's research (2019), machine learning was used to classify students' risk of failure and produced more than 90% accuracy. Asselman et al. (2023) used three ensemble models to predict students' future abilities, the results showed that XGBoost had better performance. XGBoost utilizes a boosting technique to combine predictions from multiple models, resulting in a more robust model that improves overall performance. Machine learning methods can also identify factors influencing student learning outcomes (Beckham et al., 2022). An example of the application is determining important variables that influence the GPA of IPB students (Putri et al., 2013). The mean decrease Gini random forest model obtained the importance ranking results. Random forest is an ensemble tree method built from multiple decision trees (Breiman, 2001). Each tree was created using a random sample of the training data and a random selection of features. The final prediction is made by combining the predictions of all the individual trees in the forest. The research by Putri et al. (2013) shows an error rate of almost 40% in GPA classification, and their model does not address hyperparameter tuning. The optimization method is critical to produce optimal machine learning models. Hyperparameters directly control the training process of machine learning algorithms, thus having a significant effect on model performance (Wu et al., 2019). Model-fit hyperparameter optimization techniques often demand professional knowledge and expert experience. Although there are automatic optimization techniques using default values, the accuracy of the results obtained often differs when applied to different types of problems (Yang & Shami, 2020). The methods commonly used by researchers to optimize hyperparameters are babysitting, grid search, and random search.

In principle, babysitting is a manual trial and error process, namely selecting hyperparameters based on experience or evaluating previous analysis results. The weakness of this method is that it is less appropriate for use in complex models (Steinholtz, 2018). In the grid and random search methods, parameters are tested in the model based on previously determined values. The difference is that grid search uses a complete combination of parameters, while random search only uses a few combinations chosen at random. The weakness of grid search is that it is inefficient in high-dimensional hyperparameter configuration spaces because it increases the computational process exponentially (Lorenzo et al., 2017), while random search cannot be relied on for training several complex models (Bergstra et al., 2011). For complex models or derivatives that are difficult to evaluate, Bayesian optimization is an alternative method that is more effective in searching for optimal hyperparameters. This method uses hyperparameter samples, which are later optimized based on specific criteria (Wu et al., 2019). Bayesian optimization can provide better results than grid search and random search optimization methods because Bayesian optimization can determine the quality of the experiment before starting calculations (Thornton et al., 2013). Therefore, this research aims to identify factors that influence the GPA of IPB master's program students, using random forest and XGBoost model by optimizing hyperparameters. Bayesian optimization will be used in hyperparameter tuning to increase prediction accuracy. The results obtained can be used as recommendations for the selection process for new student admissions.

## B. METHODS

The research methodology in this paper is presented by flowchart in Figure 1 with the following steps:

1. Data preparation includes data cleaning and selecting variables that will be used in creating the model.
2. The data is divided into 80% training data for building the model and 20% testing data for model evaluation.
3. Build a training model with two types of data, namely initial data and SMOTE result data, as a method for handling data imbalance.
4. Apply hyperparameter tuning using both Bayesian optimization and random search techniques. Bayesian optimization uses the first twenty initials as a sample (chosen randomly) and is refined for ten iterations, while random search optimization uses one hundred models with different hyperparameter combinations.
5. Apply cross-validation (*10-fold*) with repetition 100 times on each model, the training data is divided into ten subsets, nine parts are used as training-fold and the rest are used for test-fold. This process will be carried out in 10 iterations with random distribution of data. Cross-validation measures how well the model can generalize to data that has never been used before.
6. Determine the best model based on the evaluation of training data metrics from the optimized random forest and XGBoost models. The four metrics used are accuracy, balanced accuracy, sensitivity, and specificity.
7. Classify testing data using default and the best model from the random forest and XGBoost to see the model's goodness when applied to new data.
8. Determine the essential variables that influence the classification results of the GPA of SPs-IPB master students from the best random forest and XGBoost models.

The following is the Flowchart of the Methodology, as shown in Figure 1.

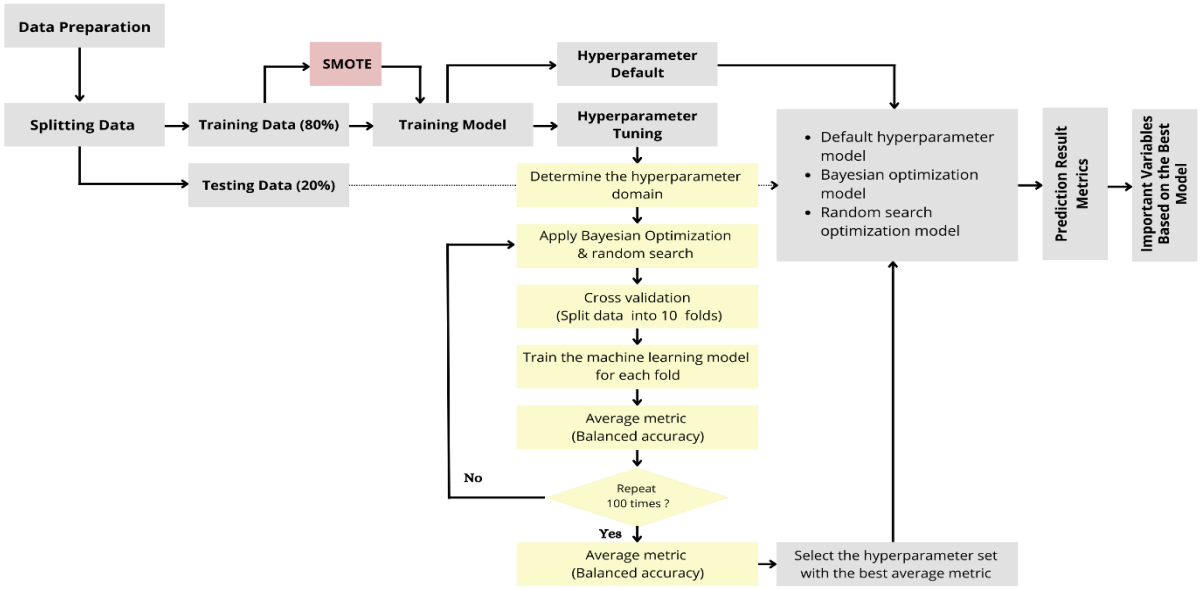


**Figure 1.** Flowchart of The Methodology

## 1. Data

The data is secondary data from IPB University Masters students in 2020 and 2021. The number of observations was 2,265 from 53 study programs, of which 53% were students from the 2020 class and 47% were students from the 2021 class. Eleven predictor variables were used to build a classification model, with information presented in Table 1. The response variable used is the GPA value in the first semester of study, the GPA value is changed into two categories, namely "Low" for students with a $GPA < 3.5$ and "High" for students with a $GPA \geq$ 3,5.

**Table 1.** List of Predictor Variable

| Variable | Name of variable | Type |
|---|---|---|
| X1 | Age | Numeric |
| X2 | Undergraduate GPA | Numeric |
| X3 | Distance undergraduate to master | Numeric |
| X4 | Gender | Category |
| X5 | Marital status | Category |
| X6 | Undergraduate university | Category |
| X7 | Undergraduate university accreditation | Category |
| X8 | Undergraduate university status | Category |
| X9 | Enrollment | Category |
| X10 | Bearer of education costs | Category |
| X11 | Study program | Category |

## 2. Random Forest

Random Forest is an ensemble model built from $n$ decision trees, as in Figure 2. In the case of classification, each tree will produce a class prediction. Most category predictions are the final result of the model. The stages of random forest algorithm consisting of $n$ observations and $m$ predictor variables are as follows (Breiman, 2001):

a. Determine the number of trees ($k$) that will be built in the random forest model.
b. Trees are created based on bootstrap using training data. Bootstrap refers to the process of random sampling replacing the original training set. By using bootstrap, each tree is trained on a subset of the observations rather than all of them. The bag is the selected subset, several trees are trained using various bags.
c. If $m_1$ is the total number of predictor variables, each node in the trees uses $m_2$ randomly selected predictor variables, where $m_2 < m_1$.
d. The above process will be repeated until $k$ trees are formed. The majority of prediction results from $k$ trees are selected as the final prediction (majority vote)

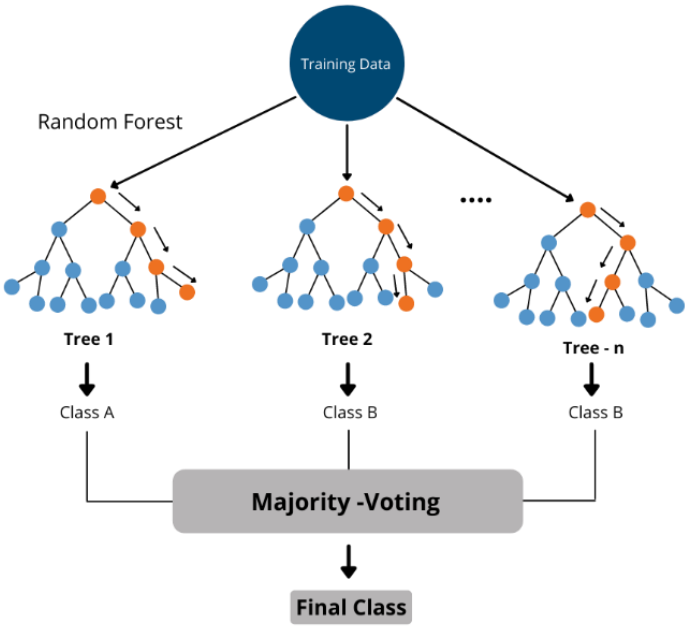


**Figure 2.** Random Forest Illustration

## 3. Extreme Gradient Boosting (XGBoost)

XGBoost is an ensemble model that combines several simple models. This combination aims to get better prediction output than using just one model. The XGBoost model was developed by Chen and Guestrin (2016). In obtaining predictions, XGBoost uses a boosting technique that will iteratively build a set of weak models on a subset of data by minimizing the mean square error score $(\hat{y} - y)$ from model $F$. This model applies gradient descent to minimize errors when creating new models. Illustration where $\hat{y}_i$ is the predicted value obtained from the following equation: XGBoost optimizes the mean square error (MSE) through a boosting process, the model is trained iteratively by giving more weight to data that is difficult to predict.

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) \tag{1}$$

$f_k$ represents a regression tree model, and $f_k(x_i)$ is the prediction score produced by the $k-th$ on the $i-th$ sample $(i = 1..n)$. The XGBoost model is optimized by searching for a new classification model that can reduce the loss function score $(l)$ in the objective function. *The loss function* is a model evaluation mechanism that measures the difference between the predicted value $\hat{y}$ and the observed $y_i$ (Wang et al., 2020). The lower the score obtained, the better the model is interpreted. The objective function is shown in Equation 2. In minimizing the loss function score, $f_t$ is added to the objective function to produce Equation 3. The addition of $f_t$ is because the model is trained additively using $\hat{y}_i^{(t)}$ in the $i-$ th prediction and $t-$ th iteration.

$$obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{t} \Omega(f_k) \tag{2}$$

$$obj^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i) + \sum_{i=1}^{t} \Omega(f_t) \tag{3}$$

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \|w\|^2 \tag{4}$$

$\Omega(f_k)$ in Equation 4 is a function that regulates model complexity to avoid overfitting. The $\gamma$ *and* $\lambda$ values are regulatory hyperparameters, while $T$ and $w$ are the number and weight of leaves in tree formation. The gain score obtained from Equation 5 is used to determine node separation. The $g_i$ dan $h_i$ values are the first and second derivatives of the loss function, respectively.

$$G = \frac{1}{2}\left[\frac{\left(\sum_{i\epsilon I_L} g_i\right)^2}{\sum_{i\epsilon I_L} h_i + \lambda} + \frac{\left(\sum_{i\epsilon I_R} g_i\right)^2}{\sum_{i\epsilon I_R} h_i + \lambda} - \frac{(\sum_{i\epsilon I} g_i)^2}{\sum_{i\epsilon I} h_i + \lambda}\right] - \gamma \tag{5}$$

## 4. Synthetic Minority Oversampling Technique (SMOTE)

SMOTE is a technique for handling data imbalance problems. Data is said to be imbalanced when there is a significant difference in proportion between the minority and majority classes (Elreedy & Atiya, 2019). The effect of this condition is that the minority class in new observations will generally be predicted as the majority class. The inability to predict the minority class will also impact the resulting accuracy. The SMOTE technique was developed by Chawla et al. (2002), with the main principle of making new synthetic observations of the minority class. It contributes to the creation of more balanced datasets, which in turn results in better-performing models that are capable of precisely projecting outcomes for both majority and minority classes. For numerical variables, synthetic observations are generated using the Euclidean distance measure as in Equation 6. Meanwhile, synthetic observations for categorical variables are generated using the Value Difference Metric (VDM) in Equation 7.

$$d(a, b) = \sqrt{\sum_{i=1}^{p_1} (a_i - b_i)^2} \tag{6}$$

$$\Delta(A, B) = \sqrt{\sum_{i=1}^{p_2} \delta(A_i, B_i)} \tag{7}$$

$a_i - b_i$ is the difference between the observed value of $a$ in the $i-$th numerical variable and the observed value of $b$ in the $i$-th numerical variable, and $p_1$ is the number of numerical variables. $\delta(A_i, B_i)$ is the distance between categories $A$ and $B$ in the $i-$th categorical variable.

## 5. Bayesian Optimization

Bayesian optimization is an iterative algorithm for finding optimal hyperparameters of machine learning models (Garrido-Merchán & Hernández-Lobato, 2020). It is a surrogate model-based optimization technique to optimize functions that are expensive to evaluate. This method determines the next hyperparameter value based on the previous results of tested hyperparameter values Aghaabbasi et al. (2023) to obtain optimal hyperparameters that approach the objective function with several iterations. The main steps of BO are as follows (Snoek et al., 2012):

a. Select an initial set of points to evaluate the objective function.
b. Use the initial points to train a surrogate model, which is a probabilistic model that approximates the actual objective function.
c. Use an acquisition function ($S$) to determine the next point to evaluate. Three commonly used acquisition functions are probability of improvement (PI), expected improvement (EI), and GP upper confidence bound (GP-UCB) (Wu et al., 2019). PI measures the probability that the objective function value at the next evaluation point will be better than the best value obtained previously. EI computes the expected degree of improvement that a point can attain when exploring the vicinity of the current optimum value. GP-UCB selects the next point to explore based on the predicted value of the Gaussian Process.
d. Update the surrogate model with the new evaluation.
e. Steps repeat the process of selecting the next point, evaluating the objective function, and updating the surrogate model until the maximum number of iterations is reached.

Commonly used probabilistic model is the Gaussian process (GP), which can easily calculate the target prediction distribution (Dewancker et al., 2016). Gaussian Process can be defined as follows (Wu et al., 2019):

$$f(x) \sim \mathrm{GP}(m(x), k(x, x')) \tag{8}$$

by assuming mean function $m(x) = 0$ and covariance function $k$ searched using Equation 9,

$$k(x_i, x_j) = \exp\left(-\frac{(x_i - x_j)^2}{2}\right) \tag{9}$$

where $x_i$ and $x_j$ represent $the\ i-\mathrm{th}$ and $j-\mathrm{th}$ sample hyperparameters. In determining the posterior distribution of the objective function $f(x)$, a matrix $K$ is constructed like Equation 10 from sample $t$ observations, and a matrix k is like Equation 11,

$$K = \begin{bmatrix} k(c, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_t) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_t) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_t, x_1) & k(x_t, x_2) & \cdots & k(x_t, x_t) \end{bmatrix} \tag{10}$$

$$\mathrm{k} = [k(x_{t+1}, x_1),\ k(x_{t+1}, x_2) \ldots\ k(x_{t+1}, x_t)] \tag{11}$$

then an average calculation is carried out-subscript $\mu_{t+1}(x_{t+1})$ and variance $\sigma^2_{t+1}(x_{t+1})$ using Equations (12) and (13).

$$\mu_{t+1}(x_{t+1}) = \mathrm{k}^T K^{-1} f_{1+t} \quad (12)$$

$$\sigma^2_{t+1}(x_{t+1}) = k(x_{t+1}, x_{t+1}) - \mathrm{k}^T K^{-1}\mathrm{k} \quad (13)$$

after getting the posterior distribution, then look for new samples by maximizing the acquisition function (S) using Equation 14,

$$x_{best} = arg \max_{x_i \epsilon x_{1:t}} f(x_i) \quad (14)$$

The process of updating the surrogate model is carried out until the point of convergence, namely the condition that the Probability of Improvement (PI) value is less than the specified *threshold* value (Lorenzo et al., 2017). The value PI can be calculated using the following equation:

$$PI = \frac{\Phi(\mu(x_{new}) - f(x_{best}))}{\sigma(x_{new})} \quad (15)$$

## C. RESULT AND DISCUSSION

### 1. Handling Imbalanced data

GPA data at the beginning of the semester is mainly in the range of 3.5 to 4, with an average of 3.68. There is a big difference between the proportions of the response variable categories. Students who received a GPA in the "High" category had a percentage of 81%, and a GPA in the "Low" category had only 19%. These differences indicate an imbalance in the data to be used. The SMOTE method will be used at the modeling stage to overcome data imbalances. Applying the SMOTE technique is expected to increase classification accuracy in the minority class. The SMOTE function used comes from the themis package with a value of $neighbors = 5$, neighbors is the number of nearest neighbors used to generate new minority class observations. Applying SMOTE to the training data increases the proportion of minority classes (Low), as illustrated in Figure 3 in red. The GPA category became balanced after adding data synthesis of around 64%.

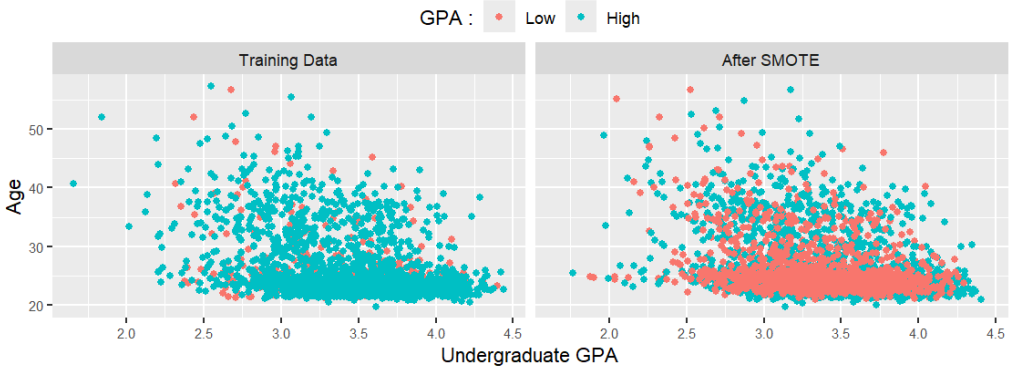


**Figure 3.** Illustration of adding synthetic data from the SMOTE technique

## 2. Hyperparameter Model

In this paper, the optimized random forest hyperparameters were the value of the trees, minimum node size, and mtry. Trees is the number of tree used to build the model with a default value of 500. Minimum node size is the minimum number of data samples to create branches at each node with a default value of 1. Mtry is the number of predictor variables ($m$) used to determine the split for each tree formed, with default value $\sqrt{m}$. The mtry hyperparameter can affect the performance of the random forest model, if the value is too small there is potential for overfitting, a value that is too large can cause the model to be too complex.

Hyperparameters optimized in XGBoost include nrounds, max depth, learning rate, gamma and sample size. Nrounds is the number of trees in model creation with a default value of 100. Max depth is the maximum depth level in a tree with a default value of 6. Learning rate is a hyperparameter to control the algorithm step size for updating model weights. Gamma is the minimum loss reduction required to create a new tree split. Sample size is the ratio of subsamples used in the training data to build the tree. If the value is set to 0.5, the tree creation uses 50% of the random data. The domain range of hyperparameter values for optimization of each model is presented in Table 2.

**Table 2.** Hyperparameter Domain

| Model | Hyperparameter | Domain | Default |
|---|---|---|---|
| Random Forest | trees | 100 - 2000 | 500 |
| | min node size | 10 - 40 | 1 |
| | mtry | 3 - 12 | $\lvert\sqrt{m}\rvert$ |
| XGBoost | nrounds | 1 - 2000 | 100 |
| | Max depth | 1 - 15 | 6 |
| | Learning rate | 0 - 1 | 0,3 |
| | Gamma | 0 - 1 | 0 |
| | Sample size | 0 - 1 | 1 |

## 3. Analysis of Random Forest and XGBoost

The results of training data evaluation metrics after hyperparameter tuning are presented in Figure 4. Before using the SMOTE technique, the average accuracy of random forest and XGBoost from the default model and optimization results was above 75%. However, the average sensitivity obtained by the model is still very low. This illustrates that the model cannot predict the minority class (low GPA) well. The implication is that the focus model predicts the majority class (high GPA), as seen from the average specificity of all models above 95%. After applying the SMOTE technique, the average sensitivity of the random forest and XGBoost models increased. This indicates that SMOTE can increase accuracy in low GPA classification.

The random forest model with Bayesian optimization performs better than the default and random search models. Bayesian optimization excels in sensitivity, specificity, and accuracy metrics, while the balanced accuracy value is relatively the same as that of the model produced by random search. The metric values produced by the XGBoost model have an extensive range, especially the accuracy, sensitivity, and specificity metrics. Therefore, XGBoost is susceptible to overfitting if it does not use hyperparameter tuning. In the XGBoost model, Bayesian and random search obtain similar average balanced accuracy values. However, Bayesian produces a higher average sensitivity, namely 68%. This shows that XGBoost with Bayesian optimization

is better at classifying low GPAs than other models. Based on the Bayesian optimization model results, the sensitivity metric of XGBoost is 25% better than random forest. However, random forest had 19% better accuracy and 30% specificity, as shown in Figure 4.

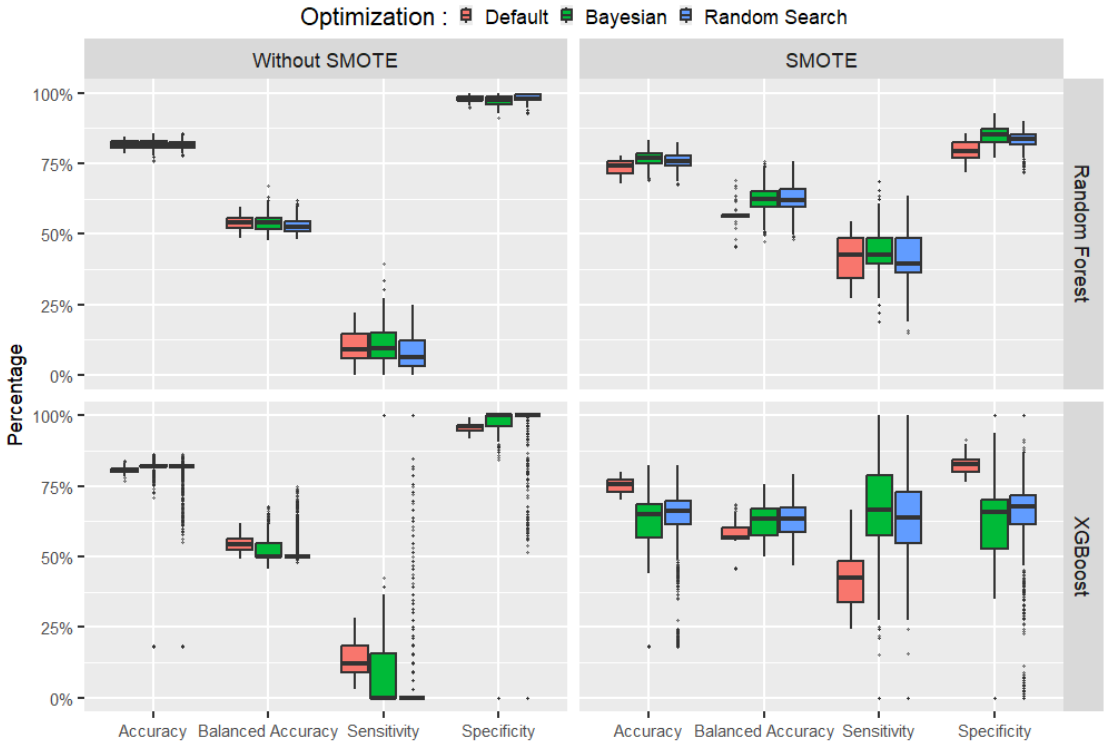


**Figure 4.** Boxplot of random forest and XGBoost model evaluation metrics

At the testing data classification stage, selecting the best model is not only based on its accuracy value. In imbalanced data case, it is necessary to consider the value of balanced accuracy because this metric helps minimize overall classification error (Thölke et al., 2023). The combination of hyperparameter values from the optimization results used in building the classification model for testing data is presented in Table 3. The hyperparameter values come from the model with the highest balanced accuracy.

**Table 3.** Best Model Hyperparameters

| Model | Hyperparameter | Default | Bayesian | Random Search |
|---|---|---|---|---|
| Random Forest | Trees | 500 | 235 | 517 |
| | Mtry | 3 | 3 | 7 |
| | Min node size | 1 | 32 | 17 |
| XGBoost | nrounds | 100 | 925 | 624 |
| | Max depth | 6 | 5 | 6 |
| | Learning rate | 0,3 | 0,001 | 0,09 |
| | Gamma | 0 | 0,00005 | 0,00001 |
| | Sample size | 1 | 0,24 | 0,4 |

The metric values of the testing data classification results are presented in Table 4, all models produce accuracy above 70%. The highest random forest accuracy was obtained from the random search model with a value of 78.36%, but the sensitivity value was less than 50%.

Bayesian optimization excels at a balanced accuracy of 68.94% and sensitivity of 54.87%. The highest specificity resulted from the random search optimization model with a value of 84.90%. The highest accuracy of XGBoost was obtained from the default model with a value of 78.36%, but the resulting sensitivity was less than 50%. The highest balanced accuracy and sensitivity were obtained from the Bayesian model, namely 70.22% and 79.26%. In general, the Bayesian optimization model can classify low GPAs well in testing data. This can be seen from the high sensitivity values in random forest and XGBoost due to Bayesian optimization. Time in the model evaluation process shows that Bayesian optimization is more effective than random search.

**Table 4.** Evaluation Metrics of Best Model

| Model | Optimization | Accuracy | Balanced Accuracy | Sensitivity | Specificity | Time (hour) |
|---|---|---|---|---|---|---|
| Random forest | Default | 0,7461 | 0,6550 | 0,5121 | 0,7978 | 0,2 |
| | Bayesian | 0,7792 | 0,6894 | 0,5487 | 0,8301 | 2 |
| | Random search | 0,7836 | 0,6684 | 0,4878 | 0,8490 | 6 |
| XGBoost | Default | 0,7814 | 0,6765 | 0,4621 | 0,8609 | 0,3 |
| | Bayesian | 0,7045 | 0,7022 | 0,7926 | 0,6618 | 2,5 |
| | Random search | 0,7792 | 0,6799 | 0,5243 | 0,8355 | 7 |

The best model resulting from Bayesian optimization is used to identify important variables. The size of the variable contribution is measured using the Mean Decreased Gini (MDG). Mean decrease Gini is a method for measuring the importance of features in a decision tree model. It calculates the average decrease in the Gini index caused by using a feature to divide nodes in a decision tree. In general, the more significant the decrease in the average Gini caused by a feature, the more important the feature is in predicting the target variable. Figure 5 displays important variables that influence the GPA classification results of SPs-IPB master students. The random forest and XGBoost models produce the same two most important variables, namely Undergraduate university status (X8) especially in the PTNBH category, and Undergraduate university (X6) especially on the undergraduate campuses located on the island of Java. Gender (X4) and Enrollment (X9) are the two variables with the least important from the random forest and XGBoost.

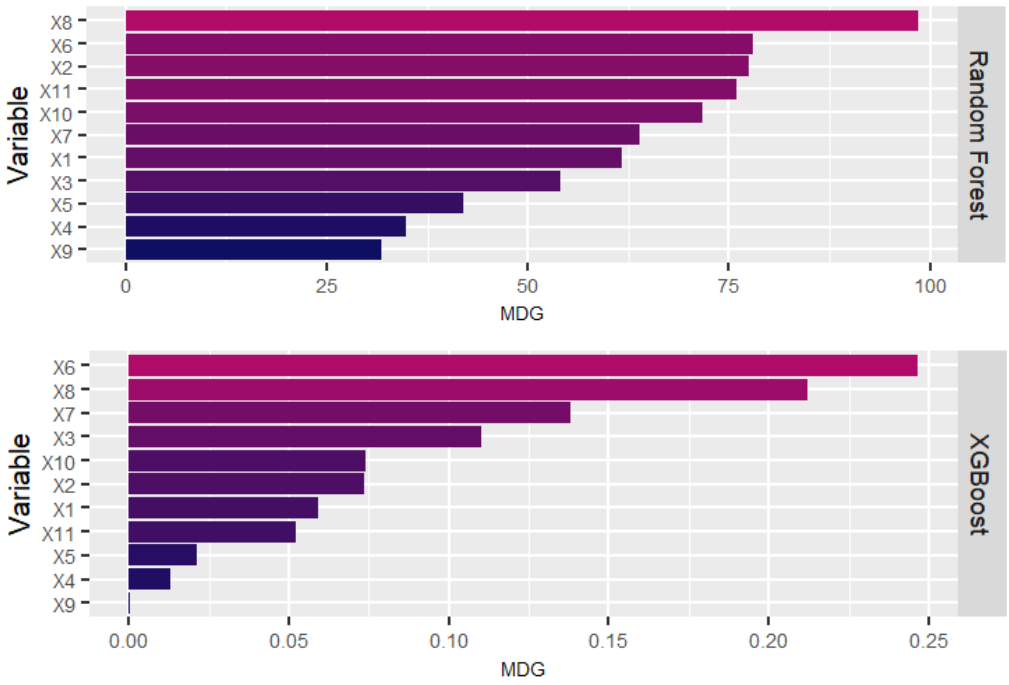


**Figure 5.** Important variables of the best random forest and XGBoost models

## D. CONCLUSION AND SUGGESTIONS

This research applies the random forest and XGBoost models to identify variables that influence the GPA of SPs-IPB master's program students. The results obtained can be used as recommendations for the selection process for new student admissions. Each model compares three hyperparameter treatments, namely default values, Bayesian optimization results, and random search optimization results. Hyperparameter optimization is used to obtain the best model with the highest balanced accuracy value. The data used is unbalanced, so it is difficult for the model to classify minority classes. The application of SMOTE at the analysis stage can increase the sensitivity value, and this metric is a measure of the model's ability to classify minority classes. The best model considering the balanced accuracy value is the Bayesian optimization model. The important variables obtained based on the Mean Decrease Gini ranking of the random forest and XGBoost models. The variables that have the most influence on students' GPA are Undergraduate University Status (X8) and Undergraduate University (X6). Meanwhile, the variables with the smallest influence are Gender (X4) and Enrollment (X9). Suggestions for developing this study could be to apply other methods in interpreting important variables. Methods that can be used include SHAP (SHapley Additive explanations). According to Nohara et al. (2022), the SHAP value is more consistent in explaining the contribution of features to the model prediction results. In the tuning process, other hyperparameters can be added to make the results more optimal.

## REFERENCES

Aghaabbasi, M., Ali, M., Jasinski, M., Leonowicz, Z., & Novak, T. (2023). On Hyperparameter Optimization of Machine Learning Methods Using a Bayesian Optimization Algorithm to Predict Work Travel Mode Choice. *IEEE Access*, *11*(January), 19762–19774. https://doi.org/10.1109/ACCESS.2023.3247448

Ahmed, S. A., & Khan, S. I. (2019). A machine learning approach to Predict the Engineering Students at

risk of dropout and factors behind: Bangladesh Perspective. *10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 1–6. https://doi.org/10.1109/ICCCNT45670.2019.8944511

Asselman, A., Khaldi, M., & Aammou, S. (2023). Enhancing the prediction of student performance based on the machine learning XGBoost algorithm. *Interactive Learning Environments*, *31*(6), 3360–3379. https://doi.org/10.1080/10494820.2021.1928235

Beckham, N. R., Akeh, L. J., Mitaart, G. N. P., & Moniaga, J. V. (2022). Determining factors that affect student performance using various machine learning methods. *Procedia Computer Science*, *216*(2022), 597–603. https://doi.org/10.1016/j.procs.2022.12.174

Beltrán-Velasco, A. I., Donoso-González, M., & Clemente-Suárez, V. J. (2021). Analysis of perceptual, psychological, and behavioral factors that affect the academic performance of education university students. *Physiology and Behavior*, *238*(June). https://doi.org/10.1016/j.physbeh.2021.113497

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011*, 1–9. https://proceedings.neurips.cc/paper_files/paper/2011/hash/86e8f7ab32cfd12577bc2619bc635690-Abstract.html

Breiman, L. (2001). Random Forests. In R. E. Schapire (Ed.), *Machine Learning* (Vol. 45, Issue 1, pp. 5–32). Kluwer Academic Publishers. https://doi.org/10.1023/A:1010933404324

Caraka, R. E., & Sugiarto, S. (2017). Path analysis of factors affecting student achievement. *Jurnal Akuntabilitas Manajemen Pendidikan*, *5*(2), 212–219. https://doi.org/10.21831/amp.v5i2.10910

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique Nitesh. *Journal of Artificial Intelligence Research*, *16*(Sept. 28), 321–357. https://arxiv.org/pdf/1106.1813.pdf%0Ahttp://www.snopes.com/horrors/insects/telamonia.asp

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-Augu*, 785–794. https://doi.org/10.1145/2939672.2939785

Cheng, S. T., & Kaplowitz, S. A. (2016). Family economic status, cultural capital, and academic achievement: The case of Taiwan. *International Journal of Educational Development*, *49*, 271–278. https://doi.org/10.1016/j.ijedudev.2016.04.002

Dewancker, I., McCourt, M., & Clark, S. (2016). *Bayesian Optimization for Machine Learning : A Practical Guidebook*. 1–15. http://arxiv.org/abs/1612.04858

Elreedy, D., & Atiya, A. F. (2019). A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance. *Information Sciences*, *505*, 32–64. https://doi.org/10.1016/j.ins.2019.07.070

Fahd, K., Venkatraman, S., Miah, S. J., & Ahmed, K. (2022). Application of machine learning in higher education to assess student academic performance, at-risk, and attrition: A meta-analysis of literature. *Education and Information Technologies*, *27*(3), 3743–3775. https://doi.org/10.1007/s10639-021-10741-7

Garrido-Merchán, E. C., & Hernández-Lobato, D. (2020). Dealing with categorical and integer-valued variables in Bayesian Optimization with Gaussian processes. *Neurocomputing*, *380*, 20–35. https://doi.org/10.1016/j.neucom.2019.11.004

Lorenzo, P. R., Nalepa, J., Kawulok, M., Ramos, L. S., & Pastor, J. R. (2017). Particle swarm optimization for hyper-parameter selection in deep neural networks. *GECCO 2017 - Proceedings of the 2017 Genetic and Evolutionary Computation Conference*, 481–488. https://doi.org/10.1145/3071178.3071208

Nohara, Y., Matsumoto, K., Soejima, H., & Nakashima, N. (2022). Explanation of machine learning models using shapley additive explanation and application for real data in hospital. *Computer Methods and Programs in Biomedicine*, *214*(February), 1–7. https://doi.org/10.1016/j.cmpb.2021.106584

Putri, S., Saefuddin, A., & Sartono, B. (2013). Identification of Affecting Factors on the GPA of First Year Students at Bogor Agricultural University Using Random Forest. *Xplore: Journal of Statistics*, *2*(1), 1–5. http://repository.ipb.ac.id/handle/123456789/65917

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, *4*, 2951–2959. https://proceedings.neurips.cc/paper/2012/hash/05311655a15b75fab86956663e1819cd-Abstract.html

Steinholtz, O. S. (2018). *A Comparative Study of Black-box Optimization Algorithms for Tuning of Hyper-parameters in Deep Neural Networks*. http://urn.kb.se/resolve?urn=urn:nbn:se:ltu:diva-69865

Thölke, P., Mantilla-Ramos, Y. J., Abdelhedi, H., Maschke, C., Dehgan, A., Harel, Y., Kemtur, A., Mekki Berrada, L., Sahraoui, M., Young, T., Bellemare Pépin, A., El Khantour, C., Landry, M., Pascarella, A., Hadid, V., Combrisson, E., O'Byrne, J., & Jerbi, K. (2023). Class imbalance should not throw you off balance: Choosing the right classifiers and performance metrics for brain decoding with imbalanced data. *NeuroImage*, *277*(April). https://doi.org/10.1016/j.neuroimage.2023.120253

Thornton, C., Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2013). Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *Part F1288*, 847–855. https://doi.org/10.1145/2487575.2487629

Wang, C., Deng, C., & Wang, S. (2020). Imbalance-XGBoost: leveraging weighted and focal losses for binary label-imbalanced classification with XGBoost. *Pattern Recognition Letters*, *136*, 190–197. https://doi.org/10.1016/j.patrec.2020.05.035

Wu, J., Chen, X. Y., Zhang, H., Xiong, L. D., Lei, H., & Deng, S. H. (2019). Hyperparameter optimization for machine learning models based on Bayesian optimization. *Journal of Electronic Science and Technology*, *17*(1), 26–40. https://doi.org/10.11989/JEST.1674-862X.80904120

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, *415*, 295–316. https://doi.org/10.1016/j.neucom.2020.07.061

Žalėnienė, I., & Pereira, P. (2021). Higher Education For Sustainability: A Global Perspective. *Geography and Sustainability*, *2*(2), 99–106. https://doi.org/10.1016/j.geosus.2021.05.001