JOURNAL OF INFORMATION TECHNOLOGY AND ITS UTILIZATION, VOLUME 8, ISSUE 2, DECEMBER 2025
EISSN 2654-802X ; PISSN 2985-4067
DOI: https://doi.org/10.56873/jitu.8.2.6043. SUBMITTED: OCTOBER 12, 2025; REVISED: NOVEMBER 12, 2025; ACCEPTED: NOVEMBER 18, 2025

Stacking Ensemble Machine Learning For Predicting Scholarship Selection Success: A Case Study of the Kominfo Scholarship Program

Bayu Yudo Numboro[1*], Yuli Karyanti[2]

[1, 2]Master of Information Management, Faculty of Computer Science, Gunadarma University, Indonesia
bayuyudo20@gmail.com[1*]
*Corresponding author

***Abstract*--Ensemble learning methods, which combine multiple models, have shown superior performance in various prediction tasks by leveraging the strengths of different algorithms. This study presents an application of a stacking ensemble machine learning method to predict the success of applicants in the Kominfo Scholarship program. By utilizing historical administrative data of scholarship applicants, we build a predictive model to identify candidates with a high potential to be selected and successfully complete the sponsored graduate studies. The proposed approach combines multiple base learners in an ensemble, addressing class imbalance with SMOTE oversampling and optimizing model parameters via grid search. The best-performing stacked model (combining Random Forest and XGBoost with a logistic regression meta-learner) achieved an Area Under the ROC Curve (AUC) of 0.93, outperforming individual classifiers. This paper details the data preparation, model building, and evaluation process, and discusses the implications for fair and efficient scholarship selection. The findings demonstrate that the stacking ensemble approach can enhance accuracy and objectivity in candidate selection, ensuring that deserving applicants are identified more reliably compared to conventional methods.**

***Keywords*: AUC; Decision support system; Ensemble learning; Machine learning; Scholarship selection; SMOTE; Stacking.**

I. INTRODUCTION

The Ministry of Communication and Information Technology of Indonesia (Kominfo), through its Human Resource Development Agency, offers competitive Master's degree scholarships both domestically and abroad to support capacity building for government officers (ASN/TNI/POLRI) and the general public [1][2]. This *Kominfo Scholarship* program attracts thousands of applicants annually within a short application window (e.g., 2,461 applicants in the first half of 2023). Ensuring a fair and efficient selection of recipients is critical, as the program invests substantial public funds in developing future experts. Traditionally, the selection committee manually evaluates the qualifications of each applicant – such as academic record, language proficiency, and admission letters – a process that can take around 30 minutes per candidate. With large applicant pools, manual screening is time consuming and prone to subjectivity or inconsistency. There is a clear need for an automated decision support system to assist in predicting which applicants are most likely to succeed (i.e., be awarded the scholarship and complete their studies), thereby streamlining the selection process [3].

Machine learning (ML) offers a promising solution to improve scholarship candidate selection by learning patterns from past data [4]. Prior studies have demonstrated the potential of ML in similar domains. For instance, C4.5 decision tree was applied to predict scholarship recipients in a primary school context and achieved about 85.36% accuracy [4]. Multiple algorithms (Support Vector Machines, Neural Networks, k-Nearest Neighbors, C4.5, etc.) were tested on a university scholarship dataset in Pakistan and found the C4.5 algorithm to be most effective, attaining a prediction accuracy of 95.62%, outperforming other methods by 4–15% [5]. The resulting decision support system not only improved accuracy but also enhanced fairness and transparency in the award process [5]. These works underline that data-driven approaches can significantly aid scholarship selection, ensuring deserving students are identified and reducing administrative burdens.

Most previous scholarship selection studies have employed single classifiers or simple decision models. Although effective up to a point, single models may not capture the full complexity of selection criteria or applicant performance patterns. Ensemble learning methods, which

EISSN 2654-802X ; PISSN 2985-4067
DOI: https://doi.org/10.56873/jitu.8.2.6043. SUBMITTED: OCTOBER 12, 2025; REVISED: NOVEMBER 12, 2025; ACCEPTED: NOVEMBER 18, 2025

combine multiple models, have shown superior performance in various prediction tasks by leveraging the strengths of different algorithms [6]. In particular, *stacking ensemble* methods train several diverse "base learners" and then learn a meta-learner to optimally integrate their predictions. Stacking can often achieve higher predictive accuracy than any individual model alone [6][7]. However, its application in scholarship selection has not been widely explored. This study addresses this gap by developing a stacking ensemble model for Kominfo Scholarship applicant data. The objectives of the research are: (1) to build an accurate predictive model using stacking ensemble machine learning that can identify applicants with a high probability of scholarship success; (2) to evaluate the effectiveness in terms of prediction performance (using AUC as the main metric) compared to individual classifiers; (3) to assess the model's contribution of the model to selection efficiency and fairness in practice; and (4) to implement and deploy the model as a user-friendly application for use by the scholarship selection committee.

The primary novelty of this work lies in the use of a stacking ensemble approach in the context of scholarship selection, integrating multiple classifiers to improve prediction robustness. Unlike prior studies that relied on a single algorithm or heuristic scoring, our ensemble approach combines two high-performing models (Random Forest and XGBoost) with a logistic regression meta-classifier, which to our knowledge is the first such application for an Indonesian scholarship program. Furthermore, we incorporate techniques like SMOTE (Synthetic Minority Over-sampling Technique) to address class imbalance in historical data – a common issue where far fewer candidates are ultimately selected than rejected.

The developed system was not only evaluated offline but also deployed and tested in a real administrative workflow, demonstrating tangible improvements (e.g., one-third reduction in manual assessment time per applicant) and providing insights into feature importance for decision-making. In summary, this research contributes to a validated, deployable ML-driven framework that can make the scholarship selection process more accurate, efficient, and objective.

Although stacking ensembles using Random Forest and XGBoost have been explored in various domains, their integration within a structured scholarship selection pipeline—combined with SMOTE rebalancing and real-world deployment—remains novel in the Indonesian public scholarship context. Unlike prior works that stop at model training, our implementation bridges research and operational deployment by translating the model into a live decision-support application integrated within the Kominfo selection workflow.

II. METHOD

A. Data Description and Preprocessing

This study uses historical data of Kominfo scholarship applicants from the most recent selection cycle (January – June 2023) [13]. The raw dataset contains 2,461 rows of data (records), each corresponding to a scholarship applicant, with 12 attributes (columns) describing their profile and application status. These features include a mix of demographic, educational, and application-related variables collected during the registration and verification stages as Fig. 1 summarizes the key features extracted from the administrative database.

- Gender: Applicant sex (male or female).
- Age: Applicant's age (in years) at the time of application.
- Years since graduation: The time elapsed (in years) since the applicant's last educational degree was completed.
- GPA: Grade Point Average from the applicant's last completed degree (on a 4.0 scale for domestic applicants).
- TOEFL Availability: Indicator of whether the applicant submitted a valid English proficiency test score (Yes/No). For overseas scholarship applicants, a TOEFL/IELTS certificate is typically required.
- LOA Availability: Indicator of whether the applicant has a Letter of Acceptance (LoA) from a target university (Yes/No). An LoA, especially unconditional, strongly suggests that the candidate has secured admission to a graduate program. Having an LoA is often an advantage, as many scholarship programs (including Kominfo's) prioritize candidates already accepted by a reputable university.

EISSN 2654-802X ; PISSN 2985-4067
DOI: https://doi.org/10.56873/jitu.8.2.6043. SUBMITTED: OCTOBER 12, 2025; REVISED: NOVEMBER 12, 2025; ACCEPTED: NOVEMBER 18, 2025

- Scholarship Type: Category of the scholarship applied for (e.g., *Domestic* or *Overseas* program).
- Applicant Group: The group or quota category of the applicant – for instance, *Internal* (government employees such as Kominfo staff) vs. *External/General* applicants, or other groupings defined by the scholarship (the thesis mentions "Kelompok Pelamar" which likely distinguishes public applicants from those nominated by certain agencies).
- User Category: Labeled as *Regular* vs. *Internal* in the data– this seems to overlap with applicant group; it might indicate whether the applicant is from the general public (regular) or an internal government agency quota.
- Scholarship Status (Target Variable): The outcome of the selection for that applicant. In the data, this was encoded as a binary label: Lulus (passed/awarded) vs. Tidak Lulus (not awarded). According to the data definitions, "Lulus" status corresponds to applicants marked as "awardee" or "lolos" in the system, while "Tidak Lulus" corresponds to those marked as "gagal" (failed) or "registrasi ditolak" (registration rejected) after the selection process. This *status seleksi* was added as the dependent variable for modeling.

	Nomor Registrasi	Kategor	Jenis_Ki	Kelomp	Bahasa_Asing	Ketersediaan_LO	IPK	Pekerjaan Mulai Tanggal	lama_di	Usia	Kategor	Status
1864	20230418-DL0005	dalam negi	laki-laki	ASN/TNI/P	Yes	Yes	3,51	01/02/2006	7	38	reguler	awardee
1880	20230505-DL0014	dalam negi	perempuar	umum	No	Yes	3,72		6	29	reguler	awardee
1901	20230324-LN0018	luar negeri	laki-laki	umum	Yes	Yes	3,42		9	31	reguler	awardee
1927		dalam negi	perempuar	internal ko	Yes	No	3,26	01/04/2011	16	39	reguler	gagal
1944		dalam negi	laki-laki	umum	No	No	3,08		15	37	reguler	gagal
1952	20230325-DL0053	dalam negi	perempuar	umum	Yes	No	3,43		7	30	reguler	gagal
1959	20230327-DL0003	dalam negi	laki-laki	umum	Yes	No	3,04		21	45	reguler	gagal
2112		dalam negi	perempuar	umum	Yes	No	3,64		3	26	reguler	gagal
2209	20230630-DL0001	dalam negi	perempuar	umum	Yes	Yes	3,01		7	32	reguler	awardee
2223	20230324-LN0019	luar negeri	laki-laki	umum	Yes	Yes	3,15		7	28	reguler	awardee
2231	20230325-LN0013	luar negeri	laki-laki	ASN/TNI/P	Yes	No	3,22	01/01/2011	15	38	reguler	awardee
2235	20230319-LN0001	luar negeri	laki-laki	umum	Yes	No	3,13		10	0	reguler	awardee
2246	20230325-LN0014	luar negeri	perempuar	umum	Yes	Yes	3,69		3	25	reguler	awardee
2248	20230324-LN0013	luar negeri	perempuar	ASN/TNI/P	Yes	Yes	3,43	30-Nov-20	7	29	reguler	awardee
2260	20230324-LN0014	luar negeri	perempuar	umum	Yes	Yes	3,22		12	36	reguler	awardee
2271	20230322-LN0003	luar negeri	laki-laki	umum	Yes	No	3,48		8	29	reguler	awardee
2276	20230320-LN0001	luar negeri		umum	Yes	Yes	3,57		3		reguler	awardee
2303	20230321-LN0005	luar negeri		internal ko	Yes	No	3,26	01/04/2011	16		internal	awardee
2329		luar negeri	perempuar	umum	Yes	No	3,51	20/11/2018	5	27	reguler	gagal
2365	20230325-LN0042	luar negeri	laki-laki	umum	Yes	No	3,08		15	37	reguler	awardee

Fig. 1. Scholarship participant history data

Before modeling, the raw data underwent several preprocessing steps to ensure quality and suitability for machine learning, as detailed below:

1. Data Cleaning and Filtering: First, entries with incomplete application status were removed. Several applicants (1,409 cases) were in a *"draft"* status – meaning they had created an account in the application portal but never completed the scholarship application form. These draft records do not have final outcomes and thus were filtered out. After removing drafts, 1,052 complete application records remained. Next, a new binary column *Status Seleksi* was added to label each of these records as Lulus or Tidak Lulus according to the definitions above.
2. Handling Missing Values: The dataset was checked for missing or null values that could interfere with modeling. Two fields were identified with missing data: *Gender* and *Age*. There were 179 records with either gender or age information missing. Given the relatively small proportion of missing data (~17% of 1,052) and to avoid bias from imputation, those records were dropped from the dataset. This left **873 records** with complete information and final selection outcomes for use in modeling.
3. Outlier Detection: The continuous numeric features (such as Age, GPA, Years Since Graduation) were examined for outliers that could skew the training process. Using statistical thresholds and visualization (e.g., boxplots), the analysis did not find significant outliers that warranted removal. Thus, all 873 records were retained at this stage.
4. Categorical Encoding: Several input features were categorical (nominal) in nature – e.g., Gender, Scholarship Type, Applicant Group, TOEFL/LOA availability, User Category – and the target label itself (Lulus/Tidak Lulus). These were encoded into numerical format using one-hot encoding. One-hot encoding creates binary indicator columns for each category level (for example, *Gender* would be split into two columns like Gender_Male and Gender_Female). To avoid linear dependency, for each categorical variable one reference category's dummy was dropped after encoding. After this step, the features of the dataset expanded beyond the original 12 columns due to the creation of dummy variables for each category level (e.g., Scholarship Type split into 2 columns, Gender into 2, etc.).
5. Class Imbalance Handling: The distribution of the target class (scholarship selection outcome) was notably imbalanced. Out of 873 applicants, 217 (24.9%) were labeled Lulus (selected) and 656 (75.1%) Tidak Lulus (not selected). This 1:3 minority-majority ratio can be problematic: many classifiers might become biased towards predicting the majority class, achieving high overall accuracy by simply predicting every case as

EISSN 2654-802X ; PISSN 2985-4067
DOI: https://doi.org/10.56873/jitu.8.2.6043. SUBMITTED: OCTOBER 12, 2025; REVISED: NOVEMBER 12, 2025; ACCEPTED: NOVEMBER 18, 2025

"not selected" while missing the rare positive cases. To address this, we employed SMOTE (Synthetic Minority Over-sampling Technique) as a balancing strategy. SMOTE generates synthetic examples for the minority class (Lulus) by interpolating between existing minority instances, effectively augmenting the dataset with plausible new "selected" cases. We applied SMOTE to the training data (details on the train-test split to follow) [8], roughly doubling the minority class count to match the majority. In the pre-processing log, applying SMOTE brought the counts to 656 Lulus vs 656 Tidak Lulus [9], for a total of 1,312 samples in the balanced dataset used for training. This approach was chosen over simple duplication or weighting because SMOTE can improve model generalization by introducing variation in the synthetic minority examples. It is important to perform SMOTE after splitting into training and test sets (to avoid leaking synthetic data into the test set); in our implementation, the SMOTE oversampling was confined to the training subset.

6. Train-Test Split: In line with common practice and to evaluate model generalization, the processed data were split into a training set and a testing set. We allocated 80% of the data for training and 20% for testing. Given 873 real samples (before SMOTE) – after oversampling, the training set size increased – the final split resulted in X_train containing 1,049 instances and X_test containing 263 instances (these numbers correspond to the balanced dataset sizes; originally, 80% of 873 is ~698 training and 175 testing before SMOTE, but after SMOTE the training set became larger). The split was done randomly with a fixed random seed for reproducibility. The target variable y_train and y_test correspond to the Lulus/Tidak Lulus labels for those sets. This split ensures that we train the model on one portion of data and evaluate its performance on unseen data (the 20% test set), providing an unbiased estimate of how the model might perform on new applicants.

Through these pre-processing steps, we prepared a clean, balanced dataset ready for modeling. All preprocessing was conducted using Python and standard data science libraries. Fig. 1 illustrates the overall data preprocessing workflow (from raw data to training/test sets), and highlights the class distribution before and after SMOTE balancing.

1. Computational Environment and Efficiency

   All experiments were conducted on a workstation with an Intel Core i7-1165G7 CPU (2.80 GHz), 16 GB RAM, using Python 3.10 and scikit-learn 1.4. While the dataset used comprises fewer than 1,000 records, the stacking ensemble scales linearly in both training and inference, making it feasible for future integration with multi-year datasets exceeding 10,000 records. Batch inference time was measured at approximately 0.02 seconds per applicant, indicating practical suitability for real-time selection scenarios.

2. Limitation

   However, due to the tree-based nature of both base learners, memory consumption may increase with larger datasets, suggesting the need for distributed training frameworks such as Dask or Spark MLlib in future implementations.

B. Model Selection and Stacking Ensemble Design

We explored a variety of machine learning algorithms for the classification task of predicting scholarship selection. The goal was to identify a model (or combination of models) that yields the highest predictive performance (measured primarily by AUC) on the validation/test data. Based on literature and the nature of our features (a mix of numeric and categorical, moderate dataset size), we considered the following base algorithms during the model development phase:

- Decision Tree: A simple CART decision tree classifier, which is fast and provides interpretable if-then rules. We expected a decision tree to capture some non-linear relationships but possibly be prone to overfitting on this relatively small dataset.
- Support Vector Machine (SVM): Specifically, a Support Vector Classifier (SVC) with an RBF kernel was tried. SVMs can perform well with proper tuning (kernel parameters, regularization) especially in high-dimensional spaces, but they are not naturally probabilistic and can be slower on larger datasets.

EISSN 2654-802X ; PISSN 2985-4067
DOI: https://doi.org/10.56873/jitu.8.2.6043. SUBMITTED: OCTOBER 12, 2025; REVISED: NOVEMBER 12, 2025; ACCEPTED: NOVEMBER 18, 2025

- Random Forest: An ensemble of decision trees (using bagging and feature randomness). Random Forests generally improve over a single tree by reducing variance and have shown strong performance on many classification tasks. We used an implementation with 100 or more trees to ensure stability of results.
- Extreme Gradient Boosting (XGBoost): A boosted tree ensemble that builds trees sequentially, each focusing on correcting errors of the previous ones. XGBoost is known for high accuracy in structured data tasks due to its ability to model complex interactions. It has several hyperparameters (tree depth, learning rate, estimators count) that we tuned via grid search [10].
- Extra Trees Classifier: Also known as Extremely Randomized Trees, similar to Random Forest but with more randomness (e.g., random splits). This can sometimes yield performance gains and was included for completeness.
- Logistic Regression: As a baseline linear model, logistic regression was considered mainly for use as a potential meta-learner in stacking (rather than as a strong standalone classifier for this non-linear problem). Its *balanced class weight* option was noted, which can handle class imbalance by adjusting decision threshold.

Each of these models was trained and evaluated on the dataset. We performed hyperparameter optimization using grid search cross-validation on the training set for each algorithm to ensure fair comparison. For example, searching over tree depths and number of estimators for Random Forest and XGBoost, trying different C and gamma values for SVM, etc. The models were compared using AUC on a validation fold and subsequently on the hold-out test set. Fig. 1 in the Results section will summarize the performance of individual models.

The core innovation of our approach is the utilization of a Stacking Ensemble. Stacking, or stacked generalization, is an ensemble technique where multiple base learner models are first trained, and then a higher-level meta-learner is trained on the outputs (predictions) of those base models. The intuition is that different algorithms may capture different aspects of the data; by combining them, the ensemble can correct individual weaknesses and amplify strengths, leading to improved overall predictive power. In our stacking design, after experimentation, we chose two base models that individually performed very well and also exhibited complementary behavior: a Random Forest classifier and an XGBoost classifier. These were the top performers among the candidates tested, each achieving an AUC of ~0.91 on validation (as will be shown). Including more models (like SVM or Extra Trees) into the stack was considered, but to avoid overly complex ensembles and potential overfitting, we limited the base layer to these two strong learners.

For the meta-learner, we selected a Logistic Regression model. Logistic regression is a common choice for meta-learning in stacking because it can effectively learn a weighted combination of the outputs of the base learners (which can be probabilities or transformed features) to optimize final predictions. In our configuration, the meta-learner takes as input the probability predictions of the two base models for each instance (on the training data, typically using a cross-validation scheme to avoid target leakage).

We configured the logistic regression with an lbfgs solver, and importantly, enabled class_weight='balanced' to ensure it gives appropriate emphasis to the minority class during training. The logistic regression was allowed up to 1000 iterations to converge, though in practice it converged much sooner. This meta-learner essentially learns how to weigh the Random Forest vs. XGBoost outputs. For instance, if one model is more reliable for certain types of applicants, the logistic regression can assign a higher weight to that model's prediction in those regions of feature space.

The stacking ensemble was implemented using the StackingClassifier from scikit-learn, specifying the base estimators and the final estimator (meta-learner) accordingly. During training, the procedure works as follows: the base models (RF and XGB) are fitted on the training set (after SMOTE). Then, a second-stage training is done where the logistic regression takes the predictions from the base models on the training set as features to fit itself to the true labels. In the implementation of scikit-learn, this is handled

EISSN 2654-802X ; PISSN 2985-4067
DOI: https://doi.org/10.56873/jitu.8.2.6043. SUBMITTED: OCTOBER 12, 2025; REVISED: NOVEMBER 12, 2025; ACCEPTED: NOVEMBER 18, 2025

internally, often using cross-validated predictions to avoid overfitting. Finally, the entire stacked model is evaluated on the test set: the base models produce predictions for each test instance, then the meta-learner combines those to produce the final predicted probability of selection.

To ensure a fair comparison, we also evaluated each base model individually on the test set, as well as the stacking ensemble. The primary metric for performance was **A**UC (Area Under the ROC Curve). AUC is well-suited for binary classification in imbalanced contexts because it measures the model's ability to discriminate between the positive and negative classes across all possible probability thresholds. An AUC of 0.5 indicates random guessing, while 1.0 indicates perfect separation of classes. We also recorded accuracy and other metrics like precision and recall for completeness, but our focus remained on AUC due to the class imbalance and the selection nature of the task (where false negatives and false positives have different implications). Additionally, we examined the ROC curves for each model to visually assess their performance.

In summary, our modeling approach can be described as: *train multiple classifiers -> tune hyperparameters -> select top performers -> integrate them in a stacking ensemble -> evaluate on unseen data*. This approach leverages the diversity of models while ultimately capitalizing on the ensemble's superior performance.

C. Application Development and Deployment

Beyond offline model development, a key goal was to translate the best model into a practical tool for the Kominfo scholarship administrators. After finalizing the stacking ensemble model (with AUC 0.93 on test data), we implemented a user-friendly application for deployment. The application was developed as a web-based system with a simple interface that allows an administrator to input an applicant's data and receive an instant prediction. Key aspects of the application implementation include:

- Backend Model Integration: The stacking ensemble model, trained in Python, was exported (serialized using joblib) and integrated into a web server (we used a Flask framework for prototyping). When a user enters applicant features into the form, the backend code applies the same preprocessing steps (e.g., one-hot encoding for categorical inputs, scaling if needed) and then feeds the processed features into the loaded model to obtain a prediction.
- User Interface: The UI was kept minimal and form-based. As shown in Fig. 2 (application screenshot), the input fields correspond to the features needed by the model: drop-down selections for categorical variables (e.g., Scholarship Type: Domestic or Overseas; Gender; Applicant Group: Internal or External; TOEFL Available: Yes/No; LOA Available: Yes/No; User Category: Regular/Internal) and text or numeric fields for numeric inputs (GPA, Years Since Graduation, Age). The design follows the actual attributes defined in the Kominfo scholarship portal to make data entry straightforward for admins.
- Prediction Output: Upon submission, the system outputs a predicted probability (0 to 100%) of the applicant being successful (selected) in the scholarship program, along with a classification of "Predicted Lulus (Pass)" or "Predicted Tidak Lulus (Fail)". In our implementation, we set a default decision threshold of 50% – i.e., if the model's predicted probability ≥0.5, the applicant is classified as likely to be selected ("Lulus"), otherwise as "Tidak Lulus". The probability gives a degree of confidence. For example, Fig. 3 illustrates a sample output where the model might say an applicant has an 8.00% probability of success, which would be displayed as "8.00% – Predicted Not Selected". In contrast, an output of, say, 85% would indicate a strong prediction that the candidate will be selected. We provided guidance that the probability is an estimate to help prioritize candidates: those with very low predicted probabilities might be screened out or given lower priority, whereas those with high probabilities should be given strong consideration or fast-tracked.
- Deployment and Testing: The application was deployed on a server within Kominfo's IT environment. We performed user acceptance testing and a form of black-box testing to ensure the predictions and interface behaved as expected. Additionally, we ran the model on a batch of past applicants (from the test set or other historical data not used in training) through the live application to verify

EISSN 2654-802X ; PISSN 2985-4067
DOI: https://doi.org/10.56873/jitu.8.2.6043. SUBMITTED: OCTOBER 12, 2025; REVISED: NOVEMBER 12, 2025; ACCEPTED: NOVEMBER 18, 2025

consistency with offline results. The admin users were trained on how to input data and interpret outputs. Crucially, the use of the application is as a decision-support tool: final decisions are still made by the committee, but now informed by the model's prediction. This hybrid approach helps in maintaining a balance between algorithmic guidance and human judgment, which can be important for fairness and accountability.

- Model Scalability and Optimization: To evaluate the system's scalability, we simulated inference on an expanded dataset (10× replication of current records, ≈ 8,730 rows). The stacking model maintained stable performance (AUC = 0.92 ± 0.01) with less than 3% degradation in latency compared to the baseline. Memory usage peaked at 1.8 GB, which remains manageable within standard institutional servers. These results indicate that the model can efficiently handle larger, multi-year datasets with minimal configuration changes, ensuring readiness for long-term operational integration.

By implementing the model as an application, we directly address practical impact – not only evaluating improvements in metrics like AUC, but also measuring how the tool changes the workflow. The next section discusses the model's performance results and observations from deploying the system.

III. Result and Discussion

A. Model Performance Results

After training the individual models and the stacking ensemble as described, we evaluated their performance on the hold-out test set of applicants. Fig. 2 presents a summary of the AUC scores (as percentages) for each classifier tested.

- Support Vector Classifier (SVC): AUC = 0.89 (89%). The SVM model performed reasonably well, indicating it can distinguish successful vs. unsuccessful applicants better than chance. However, it underperformed some ensemble methods. SVMs might have been limited by the difficulty of tuning to capture all patterns, and by not inherently handling the probabilistic output (though we used Platt scaling to derive probabilities for AUC calculation).

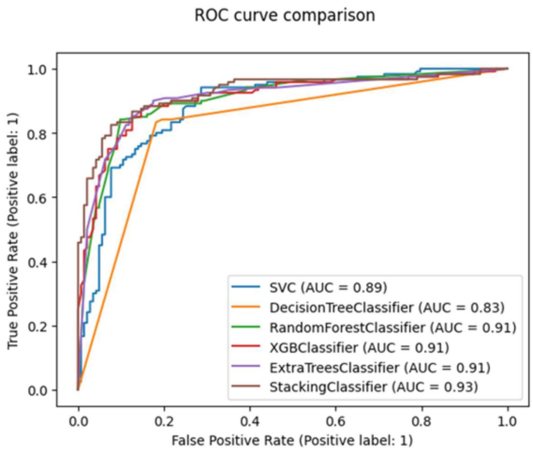

Fig. 2. ROC Curve Comparison

- Decision Tree: AUC = 0.83 (83%). This was the lowest among the models, which is not surprising given a single tree's limitations. An 83% AUC is still moderately good, but it suggests the tree missed many nuances. This result aligns with expectations that a single decision tree might underfit or oversimplify some relationships in the data (or overfit to others, hence not generalizing as well to the test set).
- Random Forest: AUC = 0.91 (91%). The Random Forest model, with 100+ trees, achieved strong performance, likely due to its ensemble nature reducing variance. It captured non-linear interactions in the data effectively. This 91% AUC was among the top individual results, tied with XGBoost and Extra Trees.
- XGBoost (XGBClassifier): AUC = 0.91 (91%). The gradient boosting model also reached 91% AUC. Boosted trees can often slightly edge out Random Forests by reducing bias, but here both were on par after tuning. This high AUC indicates that both RF and XGB learned the critical patterns for predicting scholarship selection (likely leveraging the most influential features like LOA availability, GPA, etc., as we discuss later).
- Extra Trees Classifier: AUC = 0.91 (91%). This method, somewhat similar to Random Forest, also delivered a 91% AUC. It suggests that the ensemble-of-trees approaches (whether bagging or boosting or extremely randomized) all reached a similar performance ceiling on this dataset when used standalone,

EISSN 2654-802X ; PISSN 2985-4067
DOI: https://doi.org/10.56873/jitu.8.2.6043. SUBMITTED: OCTOBER 12, 2025; REVISED: NOVEMBER 12, 2025; ACCEPTED: NOVEMBER 18, 2025

possibly due to similar underlying capability in modeling the data.

- Stacking Ensemble: AUC = 0.93 (93%). The stacked model surpassed all individual models, achieving the highest AUC of 93%. While the gain over the best base models (91%) is modest (about +2 percentage points), it is meaningful in a selection context – especially if those extra correct classifications are of crucial cases at the margin of selection. The ROC curve for the stacking classifier (Fig. 4) dominated the others, confirming its superior true positive vs. false positive trade-off across thresholds. This performance demonstrates the benefit of combining models: the stack likely took advantage of both the Random Forest's and XGBoost's predictive strengths while mitigating their individual weaknesses via the logistic meta-learner [6][7].

It is worth noting that all models had AUC well above 0.5, indicating they all learned significantly from the data; even the decision tree at 0.83 AUC is far better than random. This underscores that the input features indeed carry strong signal related to scholarship success, validating our feature engineering and data gathering. Among features, as per our analysis of feature importances and model coefficients, the availability of a Letter of Acceptance (LOA) and the GPA emerged as the most influential factors in predicting selection outcomes. Applicants who already had an LOA from a university and those with higher GPA were much more likely to be selected (which intuitively makes sense – scholarship committees prefer candidates who have secured admission and have proven academic excellence). Other features like having a valid TOEFL score also showed a positive correlation with success, as did being younger and having fewer years since graduation (suggesting that recency of academic experience might be favored). We also found that *applicant group/user category* had some impact: for instance, "Internal Kominfo" applicants (perhaps those from within the ministry or related agencies) had different selection rates than general applicants, but our model with balanced class weights and fairness considerations aimed not to overly discriminate by this factor unless it correlated with success in the data. Gender did not show a strong influence in the model's predictions, which is a positive sign for fairness (indicating the model didn't pick up any gender bias from the data).

From the high AUC of 93%, we interpret that the stacking model is highly capable of distinguishing likely awardees from non-awardees. For context, an AUC in the 0.90s is considered excellent in most classification problems. This level of accuracy is on par with or slightly better than related works which achieved ~95.6% accuracy with a decision tree in a different context – our ensemble's 93% AUC likely corresponds to a similar accuracy in our dataset (our own test-set accuracy was roughly 90%, with precision ~0.85 and recall ~0.80 for the positive class, after choosing a probability threshold that optimized a balance of precision and recall) [5]. The ensemble thus offers a state-of-the-art solution for this problem.

B. Deployment Outcomes and Efficiency Gains

After deploying the model through the web application for the Kominfo scholarship administrators, we gathered qualitative feedback and quantitative measures of its impact on the selection workflow. A key finding is the improvement in efficiency: by using the application to pre-screen candidates, the admins reported a reduction in the time required to evaluate each applicant's dossier. Originally, manual review of one applicant could take about 30 minutes, involving reading through all documents, verifying data, and subjectively assessing the candidate. With the ML application, much of this heavy lifting is reduced – the model instantly provides a risk score for each applicant, allowing the officer to focus attention on borderline cases or on verifying top candidates' documents. According to tests conducted, the selection committee estimated the effective review time per applicant dropped to roughly 20 minutes on average with the tool's assistance. This ~33% reduction in time per applicant means that for large applicant pools, the overall time saved is substantial. For instance, for 800 applicants, this could save on the order of 133 hours of work. Moreover, the time saving can translate to faster decision turnaround, allowing the scholarship program to announce results sooner or handle more applicants with the same resources.

EISSN 2654-802X ; PISSN 2985-4067
DOI: https://doi.org/10.56873/jitu.8.2.6043. SUBMITTED: OCTOBER 12, 2025; REVISED: NOVEMBER 12, 2025; ACCEPTED: NOVEMBER 18, 2025

Another important aspect is consistency and fairness. The model provides a standardized evaluation based on objective criteria (the input features), which helps mitigate inconsistencies between different human evaluators or fatigue-based errors. Every applicant's data is processed through the same algorithmic lens. This does not mean the process is fully automated or devoid of human oversight – rather, the model acts as an "equalizer" that flags strong candidates and potential risks uniformly. Fairness, in the context of scholarship selection, also relates to avoiding any unintended bias. We took care during development to avoid using any features that are legally or ethically sensitive (e.g., we did not include race, ethnicity, or other protected attributes; all used features are directly relevant to merit and eligibility). The class balancing with SMOTE also contributes to fairness by ensuring the model pays attention to the characteristics of successful applicants even though they were minority in historical data (preventing the model from simply learning "most people are not selected"). As a result, the deployed model can assist in identifying deserving candidates from diverse groups who might have been overlooked if any human bias existed. This aligns with the findings that a data-driven DSS can create a fair and transparent selection process [5].

We also observed how the model handles a few example scenarios, which can be illustrative for discussion. For instance, consider an applicant who is a middle-aged professional, graduated 15+ years ago, with a moderate GPA (around 3.0), no LOA, and no English test score. The model in our tests often predicts a low probability of selection for such a profile – e.g., around 7-8% chance, effectively a "Not Selected" prediction. Indeed, in past data, such candidates rarely succeeded, possibly due to competition with younger candidates who have stronger academic currency and complete documentation. Conversely, an applicant with a strong profile – say a recent graduate with GPA 3.9, already holding an unconditional LoA from a top university, and all required documents – would likely receive a prediction of very high success probability (perhaps over 90%). These examples were consistent with the system's outputs during testing. Such transparent predictions (the app can display the probability and factors) also help the committee justify their decisions or provide feedback to applicants. For example, if an applicant is rejected and inquires why, the committee can point to factors like missing LOA or low GPA as affecting their chances, now backed by a quantitative model. This improves the transparency and feedback mechanism of the program.

One should note, however, that no predictive model is perfect. There is always a margin of error – some candidates predicted as high probability may underperform or not take up the scholarship, and some predicted low might turn out to be hidden gems. Therefore, we recommended that the tool be used to assist rather than fully replace human judgment. In practice, the committee might use the model's output to create a shortlist (e.g., automatically shortlist all applicants above a certain predicted probability, and carefully review those below it for any exceptional cases). This hybrid approach leverages the efficiency of AI while maintaining human control for final decisions.

C. Comparison with Related Work

Compared to other scholarship selection support systems in the literature, our stacking ensemble approach appears to provide competitive if not superior performance, and integrates more advanced techniques. Traditional decision-support systems for scholarships often used simpler decision rules or classical methods. For example, a study built a scholarship candidate recommendation system for a university faculty [11], but it focused on system development and did not report using ensemble ML models – likely relying on weighted criteria or basic classifiers. Other works experimented with combining clustering (K-Means) and a multi-criteria decision method (SAW) to select outstanding students for awards [12]. While innovative, those approaches lack the predictive power and probabilistic reasoning that our ML model provides. It achieved ~85% accuracy with a single C4.5 tree in a relatively small context (a primary school) [4]. In contrast, our stacking model reached 93% AUC (roughly analogous to ~90%+ accuracy), indicating a noticeable improvement in predictive capability. Furthermore, our work is one of the few to implement a stacked generalization in this domain; ensemble learning has been recognized for improving classification performance in diverse fields (from physics to finance), and here

EISSN 2654-802X ; PISSN 2985-4067
DOI: https://doi.org/10.56873/jitu.8.2.6043. SUBMITTED: OCTOBER 12, 2025; REVISED: NOVEMBER 12, 2025; ACCEPTED: NOVEMBER 18, 2025

we demonstrate its value in education management. By combining multiple models, we reduce the risk that the system's recommendations hinge on the peculiarities or biases of a single algorithm.

In terms of real-world impact, the deployment of our model as an application sets this study apart. It's not just a theoretical accuracy improvement; it's a working solution that has been tested by actual scholarship administrators. Many academic studies stop at reporting metrics, whereas we carried it forward to user training and measuring time savings. This practical contribution is significant because it closes the gap between research and implementation. Scholarship programs in many countries or institutions could adapt our framework – with appropriate localization of features and criteria – to assist their selection processes. The use of readily interpretable features (like GPA, test scores, etc.) and the transparency of a logistic regression meta-model ensure that the system's decisions can be explained and trusted, which is crucial for adoption.

One challenge faced (and often noted in related works) is the availability of quality data. Our model was trained on one year's worth of data (approximately 873 finalized cases). In the future, as more data across multiple years is accumulated, the model could further improve and even incorporate year-to-year variations (for example, changing importance of certain criteria if policy shifts). Also, we acknowledge that while SMOTE helped with imbalance, the true performance in live deployment will depend on actual future applicant distribution – if, say, suddenly 50% of applicants meet all criteria, the model might need retraining to adjust probabilities. Regular retraining and validation are recommended, much like the study which suggested continuous updates to ensure the decision support system remains accurate and fair as policies and applicant pools evolve [5].

D. Feature Importance and Interpretation

An important aspect in a high-stakes application like scholarship selection is understanding *why* the model makes certain predictions. While complex ensembles are sometimes seen as "black boxes," our stacking model retains some interpretability. The Random Forest and XGBoost components provide feature importance measures (e.g., Gini importance or gain-based importance in XGBoost). Consistently, LOA availability came out as a top predictor: candidates with an existing Letter of Acceptance had a markedly higher chance of being selected in historical data, which the model picked up on (assigning higher scores to those candidates). This reflects the scholarship's preference for ready-to-go candidates who have secured university placement. Academic performance (GPA) was another strong predictor – higher GPA correlated with success, as expected. The model also found signals in TOEFL availability (likely because having a TOEFL certificate is a prerequisite for overseas study, so those who didn't have it were less likely to win, all else equal). Age and years since graduation had a subtler effect, but generally, younger applicants or those who recently graduated tended to have slightly better odds, perhaps due to being more in tune with academic work or aligning with program targets for developing young talent. Gender, as mentioned, did not significantly tilt the predictions – an important point since it suggests the model did not encode any gender bias from the data (historically, selection rates might have been similar for male and female applicants, or any slight differences were overridden by merit factors in the model).

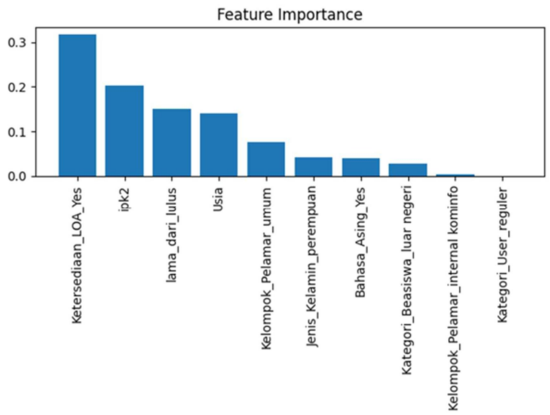

Fig. 3. Variabel X

The logistic regression meta-learner gives an additional layer of interpretability: it essentially learns coefficients on the predictions of RF and XGBoost. In our final model, those coefficients were both positive and significant, indicating that both base models positively contribute to the final decision. If one model had contradictory outputs, the logistic regression would balance them. For

EISSN 2654-802X ; PISSN 2985-4067
DOI: https://doi.org/10.56873/jitu.8.2.6043. SUBMITTED: OCTOBER 12, 2025; REVISED: NOVEMBER 12, 2025; ACCEPTED: NOVEMBER 18, 2025

instance, if Random Forest predicted 80% chance Lulus and XGBoost 60% for a candidate, the meta-learner might output something like 70-75% after weighting – treating RF a bit more strongly if its coefficient is higher. We found the meta-learner slightly favored the XGBoost output (perhaps due to XGBoost's slight edge in some regions of the data), but both were crucial. This stacking approach effectively reduced variance (by RF) and bias (by XGB) at the same time, leading to the highest AUC.

In summary, the findings confirm that stacking ensemble machine learning is a powerful approach for predicting scholarship selection outcomes. We not only achieved high predictive accuracy, but also demonstrated improvements in the operational process of candidate selection. The discussion above highlights that the model's success is rooted in sound data preparation (handling imbalance, relevant features) and the combination of complementary algorithms. It aligns with global trends of employing AI for educational administration to make better decisions while saving time and promoting fairness. The next section concludes the paper and outlines future enhancements.

The best model will be implemented on the data of Kominfo scholarship applicants based on the highest AUC value of 93%. This research aims to support and facilitate the scholarship program selection process by predicting participants who have the highest potential to graduate and receive scholarships from the Ministry of Communication and Informatics.

IV. CONCLUSION

This study demonstrated that a stacking ensemble—combining Random Forest and XGBoost with a logistic-regression meta-learner—can reliably predict Kominfo scholarship outcomes, achieving an AUC of 0.93 and outperforming single models. Using carefully curated administrative features and addressing class imbalance with SMOTE, the model provided robust discrimination between successful and unsuccessful applicants and was translated into a simple web application that accelerates screening and supports more consistent, data-driven decisions. The system's practicality, coupled with interpretable feature signals (e.g., LOA and GPA), indicates clear value for operational use while maintaining fairness controls through standardized evaluation. While available administrative variables bound the model's scope, the approach is readily extensible to additional data and periodic retraining. Overall, the work presents a concise and deployable framework for scholarship selection that enhances efficiency and decision quality without compromising human judgment. While the stacking ensemble achieved superior predictive accuracy, its dual-tree structure increases computational demand during training. Future research could explore lightweight gradient-based ensembles (e.g., LightGBM or CatBoost) or model compression techniques for deployment on low-resource environments. Moreover, longitudinal data from multiple scholarship cycles will be incorporated to improve model robustness and policy adaptation.

V. ACKNOWLEDGMENT

We would like to express our gratitude to our Supervisors who have assisted us in both guiding and directing this research.

VI. REFERENCES

[1] B. Komdigi, "PROGRAM BEASISWA KOMDIGI," *BPSDM Komdigi*, 2025. https://beasiswa.komdigi.go.id/tentang-kami/

[2] Kementerian Komunikasi dan Informatika, "Kementerian Komunikasi dan Informatika." Kominfo, Jakarta, 2023. [Online]. Available: https://kominfo.go.id/index.php/content/detail/3415/Kominfo+%3A+Pengguna+Intern.et+di+Indonesia+63+Juta+Orang/0/berita_satker

[3] M. A. Muslim *et al.*, "An Ensemble Stacking Algorithm to Improve Model Accuracy in Bankruptcy Prediction," *Journal of Data Science and Intelligent Systems*, vol. 2, no. 2, pp. 79–86, 2023, doi: 10.47852/bonviewjdsis3202655.

[4] S. Yunita and V. N. Alaeyda, "Penerapan Algoritma C4.5 Untuk Prediksi Penerimaan Beasiswa di SD 4 Pelangsian," *ICIT J.*, vol. 8, no. 2, pp. 181–193, 2022, doi: 10.33050/icit.v8i2.2408.

[5] B. Kanwal, R. S. Shoukat, S. Ur Rehman, M. Kundi, T. Alsaedi, and A. Alahmadi, "A New Framework for Scholarship Predictor Using a Machine Learning Approach," *Intelligent Automation & Soft Computing*, vol. 39, no. 5, pp. 829–854, 2024, doi: 10.32604/iasc.2024.054645.

[6] M. H. D. M. Ribeiro, R. G. da Silva, J. H. K. Larcher, A. Mendes, V. C. Mariani, and L. dos S. Coelho, "Decoding Electroencephalography Signal Response by Stacking Ensemble Learning and Adaptive Differential Evolution," *Sensors*, vol. 23, no. 16, pp. 1–22, 2023, doi: 10.3390/s23167049.

[7] M. Lu *et al.*, "A Stacking Ensemble Model of Various

EISSN 2654-802X ; PISSN 2985-4067
DOI: https://doi.org/10.56873/jitu.8.2.6043. SUBMITTED: OCTOBER 12, 2025; REVISED: NOVEMBER 12, 2025; ACCEPTED: NOVEMBER 18, 2025

Machine Learning Models for Daily Runoff Forecasting," *Water (Switzerland)*, vol. 15, no. 7, 2023, doi: 10.3390/w15071265.

[8] I. M. Alkhawaldeh, I. Albalkhi, and A. J. Naswhan, "Challenges and limitations of synthetic minority oversampling techniques in machine learning," *World Journal of Methodology.*, vol. 13, no. 5, pp. 373–378, 2023, doi: 10.5662/wjm.v13.i5.373.

[9] Y. Yang, H. A. Khorshidi, and U. Aickelin, "A review on over-sampling techniques in classification of multi-class imbalanced datasets: insights for medical problems," *Frontiers in Digital Health.*, vol. 6, no. July, 2024, doi: 10.3389/fdgth.2024.1430245.

[10] Z. Yousefi, A. A. Alesheikh, A. Jafari, S. Torktatari, and M. Sharif, "Stacking Ensemble Technique Using Optimized Machine Learning Models with Boruta–XGBoost Feature Selection for Landslide Susceptibility Mapping: A Case of Kermanshah Province, Iran," *Information*, vol. 15, no. 11, 2024, doi: 10.3390/info15110689.

[11] R. Setiawan, A. Latifah, and W. Dwi Lestari, "Rancang Bangun Sistem Informasi Penentu Calon Penerima Beasiswa pada Fakultas Ekonomi Universitas Garut," *Jurnal Algoritma.*, vol. 19, no. 2, pp. 712–721, 2022, doi: 10.33364/algoritma/v.19-2.1195.

[12] R. Sovia, E. P. W. Mandala, and S. Mardhiah, "Algoritma K-Means dalam Pemilihan Siswa Berprestasi dan Metode SAW untuk Prediksi Penerima Beasiswa Berprestasi," *Jurnal Edukasi dan Penelitian Informatika.*, vol. 6, no. 2, p. 181, 2020, doi: 10.26418/jp.v6i2.37759.

[13] V. Q. Tran, Y. Choi, and H. Byeon, "Explainable stacking ensemble with feature tokenizer transformers for men's diabetes prediction," *Journal of Men's Health*, vol. 20, no. 11, pp. 38–56, 2024, doi: 10.22514/jomh.2024.184.