

Prediksi Dropout Mahasiswa menggunakan XGBoost dengan Seleksi Fitur Boruta, Optimasi Hyperparameter, dan Interpretasi SHAP

Student Dropout Prediction using XGBoost with Boruta Feature Selection, Hyperparameter Optimization, and SHAP Interpretation

¹Silvia Maulidia Mirosa*, ²Muhammad Iqbal Akbar

^{1,2}Program Studi Teknik Informatika, Fakultas Teknik, Universitas Negeri Malang

^{1,2}Jalan Semarang 5, Malang 65145, Jawa Timur

*e-mail: silvia.maulidia.2305356@students.um.ac.id

(received: 6 October 2026, revised: 20 October 2026, accepted: 21 October 2026)

Abstrak

Prediksi mahasiswa yang berisiko *dropout* dapat membantu perguruan tinggi mengidentifikasi pola yang berkaitan dengan status studi mahasiswa. Penelitian ini mengembangkan model klasifikasi status mahasiswa (*Graduate* dan *Dropout*) menggunakan XGBoost dengan seleksi fitur Boruta, optimasi *hyperparameter* melalui GridSearchCV, dan interpretasi SHAP pada dataset *Predict Students' Dropout and Academic Success* dari UCI (3.630 data dan 36 variabel prediktor setelah kelas *Enrolled* dikeluarkan). Tiga konfigurasi dibandingkan, yaitu XGBoost *baseline*, Boruta + XGBoost, dan Boruta + GridSearchCV + XGBoost. Boruta mengurangi jumlah fitur dari 36 menjadi 11 atau sebesar 69,44%. Model Boruta + GridSearchCV + XGBoost mencapai *accuracy* 91,74% dan F1-score 89,21%, setara dengan model *baseline* (*accuracy* 91,74%; F1-score 89,36%), dan uji McNemar tidak menemukan perbedaan yang signifikan antar-model ($p > 0,05$). SHAP mengidentifikasi *Curricular units 2nd sem (approved)*, *Course*, dan *Tuition fees up to date* sebagai fitur dengan kontribusi terbesar. *Accuracy* turun menjadi 90,22% tanpa fitur semester kedua dan menjadi 74,93% tanpa seluruh fitur akademik *Curricular units*.

Kata kunci: seleksi fitur Boruta, GridSearchCV, SHAP, prediksi *dropout* mahasiswa, XGBoost

Abstract

Predicting students at risk of dropping out can help higher education institutions identify patterns associated with students' academic status. This study develops a student status classification model (*Graduate* and *Dropout*) using XGBoost, Boruta feature selection, hyperparameter optimization through GridSearchCV, and SHAP-based interpretation. The study uses the *Predict Students' Dropout and Academic Success* dataset from the UCI Machine Learning Repository, comprising 3,630 records and 36 predictor variables after excluding the *Enrolled* class. Three model configurations were compared: baseline XGBoost, Boruta + XGBoost, and Boruta + GridSearchCV + XGBoost. Boruta reduced the number of features from 36 to 11, representing a reduction of 69.44%. The Boruta + GridSearchCV + XGBoost model achieved an accuracy of 91.74% and an F1-score of 89.21%, comparable to the baseline model (accuracy: 91.74%; F1-score: 89.36%). The McNemar test revealed no statistically significant differences between the models ($p > 0.05$). SHAP identified *Curricular units 2nd sem (approved)*, *Course*, and *Tuition fees up to date* as the features with the greatest contributions to the model's predictions. Accuracy decreased to 90.22% when the second-semester feature was removed and to 74.93% when all academic *Curricular units* features were excluded.

Keywords: Boruta feature selection, GridSearchCV, SHAP, student dropout prediction, XGBoost

1 Pendahuluan

Prediksi mahasiswa yang berisiko mengalami *dropout* merupakan salah satu penerapan *machine learning* dalam bidang pendidikan yang dapat membantu perguruan tinggi mengidentifikasi pola yang berkaitan dengan status studi mahasiswa. Data akademik dan karakteristik mahasiswa dapat

<http://sistemasi.ftik.unisi.ac.id>

dimanfaatkan untuk membangun model klasifikasi status akademik. Penerapan *machine learning* untuk memprediksi *dropout* telah dilakukan pada berbagai tahap studi menggunakan data akademik [1], termasuk dengan membandingkan Random Forest dan XGBoost [2]. Salah satu dataset yang digunakan dalam penelitian semacam ini adalah *Predict Students' Dropout and Academic Success* dari UCI *Machine Learning Repository*, yang terdiri atas 4.424 data mahasiswa dengan 36 variabel prediktor demografis, sosial-ekonomi, dan akademik serta tiga kategori target, yaitu *Dropout*, *Enrolled*, dan *Graduate* [3].

Salah satu algoritma yang dapat digunakan untuk klasifikasi tersebut adalah Extreme Gradient Boosting (XGBoost), metode *tree boosting* yang dirancang untuk menghasilkan model prediksi yang efektif dan efisien [4]. Carballo-Mendivil dkk. [5] mengembangkan sistem peringatan dini *dropout* berbasis XGBoost menggunakan data pra-pendaftaran, sedangkan Praptomo dkk. [6] menerapkan XGBoost dan SHAP pada dataset mahasiswa Portugal yang sama dengan mempertahankan tiga kategori status akademik. Kedua studi tersebut berfokus pada algoritma dan data masukan, sedangkan kualitas fitur, konfigurasi model, dan penjelasan keputusan model juga perlu diperhatikan.

Dari sisi fitur, Boruta merupakan metode seleksi fitur yang membandingkan kepentingan fitur asli dengan fitur bayangan (*shadow features*) untuk mengidentifikasi fitur yang relevan terhadap target [7]. Mincevičius [8] menggunakan dataset yang sama dan membahas faktor akademik serta karakteristik mahasiswa yang berkontribusi terhadap prediksi status mahasiswa. Kinerja model juga dipengaruhi oleh konfigurasi *hyperparameter*, yang dapat dioptimalkan dengan GridSearchCV dari pustaka scikit-learn [9]. Untuk menjelaskan keputusan model, SHAP (SHapley Additive exPlanations) sebagai metode *Explainable Artificial Intelligence* (XAI) memberikan nilai atribusi kontribusi fitur terhadap prediksi [10], dan TreeSHAP memungkinkan perhitungannya secara efisien dan eksak pada model berbasis pohon [11]. Nti dan Ramanayake [12] menerapkan *explainable machine learning* untuk prediksi *dropout* guna memahami faktor yang berkaitan dengan hasil prediksi dan mendukung perancangan intervensi.

Penelitian terdahulu umumnya membahas seleksi fitur, optimasi model, atau interpretasi secara terpisah, sehingga pengaruh ketiganya terhadap performa dan efisiensi XGBoost pada klasifikasi biner belum banyak dievaluasi dalam satu alur. Tujuan penelitian ini adalah mengevaluasi pengaruh seleksi fitur Boruta dan optimasi *hyperparameter* terhadap performa XGBoost dalam memprediksi status *Graduate* dan *Dropout*. Evaluasi mencakup pengurangan fitur, perbandingan performa dan waktu komputasi tiga konfigurasi model disertai uji McNemar, analisis kontribusi fitur dengan SHAP, serta perubahan performa ketika informasi akademik semester kedua dan seluruh fitur akademik tidak tersedia. Kelas *Enrolled* tidak diikutsertakan agar klasifikasi terfokus pada dua kelas tersebut.

Hasil penelitian diharapkan bermanfaat bagi perguruan tinggi sebagai bahan pertimbangan dalam mengidentifikasi mahasiswa yang berisiko *dropout*, karena analisis SHAP dan skenario ketersediaan fitur memberikan gambaran fitur yang paling berkontribusi serta tahap studi ketika model mulai dapat digunakan. Secara akademik, penelitian ini memberikan bukti empiris mengenai pengaruh seleksi fitur dan optimasi *hyperparameter* terhadap performa dan efisiensi komputasi XGBoost, beserta alur evaluasi yang dapat direproduksi pada dataset yang sama dan dapat menjadi rujukan bagi penelitian prediksi *dropout* selanjutnya.

2 Tinjauan Literatur

Prediksi *dropout* mahasiswa merupakan salah satu topik dalam penerapan *machine learning* di bidang pendidikan tinggi. Tinjauan sistematis terhadap penelitian *university dropout* periode 2012–2024 oleh Quimiz-Moreira dkk. [13] mengidentifikasi 520 faktor yang dikelompokkan ke dalam lima kategori, yaitu demografis, sosial-ekonomi, institusional, personal, dan akademik, serta 83 model *machine learning* dengan *decision tree*, *logistic regression*, dan *random forest* sebagai yang paling banyak digunakan. SHAP dan LIME merupakan metode XAI yang paling banyak digunakan. Temuan ini memperlihatkan beragamnya faktor dan algoritma dalam prediksi *dropout* serta semakin pentingnya aspek interpretabilitas.

Goran dkk. [14] mengembangkan pendekatan prediksi *dropout* yang menggabungkan optimasi metaheuristik dengan AdaBoost dan XGBoost. Modifikasi *Sinh Cosh Optimizer* (SCHO) digunakan untuk optimasi *hyperparameter*, dan model dievaluasi pada skenario klasifikasi biner dan *multiclass*. Hasilnya mengindikasikan bahwa optimasi *hyperparameter* dapat meningkatkan performa model, dan

<http://sistemasi.ftik.unisi.ac.id>

teknik XAI diintegrasikan untuk memahami hasil prediksi. Berbeda dengan optimasi metaheuristik tersebut, penelitian ini menggunakan GridSearchCV pada ruang parameter yang telah ditentukan.

Dalam seleksi fitur, Priyatno dkk. [15] menerapkan seleksi bertahap yang terdiri atas *Variance Threshold*, *Mutual Information*, dan Boruta pada dataset mahasiswa dengan 4.423 data. Boruta mengonfirmasi 18 fitur, dan model XGBoost dengan fitur terpilih menghasilkan *accuracy* 90,77%, *precision* 92,07%, *recall* 83,68%, dan F1-score 87,63%.

Alzabidi dkk. [16] mengusulkan SOD-FE (*Supervised Outlier Detection and Feature Engineering*) yang menggabungkan deteksi *outlier* berbasis *interquartile range* (IQR), *mutual information*, korelasi Pearson, dan seleksi fitur. Pendekatan ini diuji pada dataset mahasiswa Portugal dan Slovakia dengan beberapa algoritma klasifikasi, termasuk Random Forest dan XGBoost, dan memakai SHAP untuk menjelaskan fitur yang berpengaruh. Hasilnya mengindikasikan bahwa *preprocessing*, deteksi *outlier*, *feature engineering*, dan seleksi fitur dapat menjadi bagian penting dalam pengembangan model prediksi *dropout*.

Dalam optimasi *hyperparameter*, GridSearchCV mencari secara sistematis seluruh kombinasi parameter yang ditentukan. Dewi dkk. [17] membandingkan *grid search* dan *random search* dan menunjukkan bahwa *random search* dapat menemukan konfigurasi yang baik dengan komputasi yang lebih efisien pada kondisi tertentu. Meskipun demikian, *grid search* tetap dapat digunakan ketika ruang pencarian dibatasi pada jumlah kombinasi yang terukur, sebagaimana diterapkan pada Subbab 3.4.

Dari sisi interpretasi, SHAP tercatat sebagai salah satu metode XAI yang paling banyak digunakan pada prediksi *dropout* [13], dan XAI telah diintegrasikan dengan model prediksi untuk memahami faktor yang berkontribusi terhadap hasil prediksi [14], [16].

Tabel 1 Perbandingan dengan penelitian terdahulu

Aspek	Priyatno dkk. [15]	Alzabidi dkk. [16]	Goran dkk. [14]	Penelitian Ini
Fokus	Seleksi fitur bertahap untuk prediksi <i>dropout</i>	Deteksi <i>outlier</i> dan <i>feature engineering</i> untuk prediksi <i>dropout</i>	Optimasi metaheuristik dan XAI untuk prediksi <i>dropout</i>	Klasifikasi <i>Graduate</i> dan <i>Dropout</i>
Dataset	4.423 data mahasiswa (UCI <i>Student Dropout and Academic Success</i>)	Dataset Portugal (4.424 data) dan Slovakia (261 data)	UCI <i>Student Dropout and Academic Success</i>	UCI <i>Predict Students' Dropout and Academic Success</i> , 3.630 data setelah kelas <i>Enrolled</i> dikeluarkan
Algoritma	XGBoost	Lima algoritma klasifikasi: LR, SVM, RF, XGB, MLP	AdaBoost dan XGBoost	XGBoost
Seleksi fitur	<i>Variance Threshold</i> , <i>Mutual Information</i> , Boruta	<i>Mutual Information</i> , korelasi Pearson (tahap SOD), serta RFI dan RFE (tahap <i>feature engineering</i>)	Tidak menjadi fokus utama	Boruta
Optimasi	Tidak menggunakan optimasi <i>hyperparameter</i> (parameter default)	Tidak menjadi fokus utama	Metaheuristik (<i>modified SCHO/AGbSCHO</i>) untuk <i>tuning hyperparameter</i>	GridSearchCV
Interpretasi	Tidak menjadi fokus utama	SHAP	SHAP dan SAGE	SHAP <i>TreeExplainer</i>
Validasi	<i>Stratified 10-Fold Cross-Validation</i>	<i>Hold-out</i> (80:20) dan <i>5-Fold Cross-</i>	<i>Train-test split</i> 70:30 dengan 30	<i>Hold-out</i> 80:20 <i>stratified</i> ; <i>Stratified</i>

		Validation	run independen; validasi statistik (Shapiro-Wilk, Wilcoxon signed- rank test)	5-Fold CV pada GridSearchCV; McNemar's test
Evaluasi	accuracy, precision, recall, F1-score	accuracy, precision, recall, F1-score, AUC	accuracy, precision, recall, F1-score, Cohen's Kappa, MCC	accuracy, precision, recall, F1-score, ROC- AUC, jumlah fitur, waktu fitting dan prediksi

Tabel 1 memperlihatkan bahwa penelitian terdahulu [14]–[16] membahas seleksi fitur, optimasi model, atau interpretasi dengan pendekatan berbeda, tetapi belum memadukan ketiganya dengan uji signifikansi statistik antar-model dan analisis efisiensi komputasi pada klasifikasi biner *Graduate* dan *Dropout*.

3 Metode Penelitian

Studi dilakukan dengan pendekatan eksperimen kuantitatif untuk klasifikasi status mahasiswa. Tahapan penelitian terdiri atas pengumpulan dataset, *preprocessing*, seleksi fitur Boruta, optimasi *hyperparameter*, evaluasi model, interpretasi SHAP, dan analisis skenario ketersediaan fitur, yang diuraikan pada Subbab 3.1 sampai 3.7.

3.1. Dataset

Dataset yang digunakan adalah *Predict Students' Dropout and Academic Success* dari UCI *Machine Learning Repository*. Setelah kelas *Enrolled* dikeluarkan, diperoleh 3.630 data yang terdiri atas 2.209 mahasiswa *Graduate* dan 1.421 mahasiswa *Dropout*, dengan 36 variabel prediktor. Jumlah kedua kelas tidak sama (*Graduate* sekitar 61%), sehingga pembagian data pada Subbab 3.2 dilakukan secara *stratified*. Kelas *Graduate* diberi label 0, sedangkan *Dropout* diberi label 1 sebagai kelas positif, karena kelas *Dropout* adalah kelas yang ingin dideteksi oleh model.

3.2. Preprocessing

Preprocessing diawali dengan pemeriksaan struktur data, tipe data, dan nilai kosong, kemudian variabel target dipisahkan dari variabel prediktor. Seluruh variabel telah berbentuk numerik, termasuk variabel kategorikal berkode numerik seperti *Course*, yang diperlakukan sebagai fitur nominal dan tidak diinterpretasikan berdasarkan urutan nilai kodenya. *Encoding* dengan `get_dummies()` (`drop_first=True`) tetap diterapkan sehingga jumlah fitur tetap 36. Dataset kemudian dibagi dengan proporsi 80:20 menggunakan `train_test_split` (`stratify=y`, `random_state=42`), menghasilkan 2.904 data latih dan 726 data uji, dengan pembagian *stratified* untuk mempertahankan proporsi kelas pada kedua subset. Untuk mencegah *data leakage*, yaitu penggunaan informasi yang seharusnya tidak tersedia selama pembentukan model [18], seluruh seleksi fitur Boruta dan pencarian *hyperparameter* GridSearchCV dilakukan hanya pada data latih, sedangkan data uji hanya digunakan untuk evaluasi akhir.

3.3. Seleksi Fitur Boruta

Seleksi fitur dilakukan menggunakan Boruta dengan RandomForestClassifier sebagai estimator. Random Forest merupakan metode *ensemble* berbasis *decision tree* yang menghasilkan ukuran kepentingan variabel (*variable importance*), sehingga dapat digunakan sebagai estimator seleksi fitur [19]. Estimator dikonfigurasi dengan `class_weight="balanced"` untuk menyesuaikan bobot kelas yang tidak seimbang, `max_depth=None`, dan `random_state=42`. Jumlah estimator pada Random Forest tidak ditetapkan tetap. BorutaPy dijalankan dengan `n_estimators="auto"`, sehingga algoritma menyesuaikan jumlah estimator secara otomatis pada setiap iterasi berdasarkan dimensi data. Parameter Boruta yang lain adalah `max_iter=100` sebagai batas iterasi, `perc=100` sebagai persentil ambang kepentingan fitur

bayangan, dan $\alpha=0,05$ sebagai tingkat signifikansi. Seleksi menghasilkan 11 fitur berstatus *confirmed*, yang kemudian digunakan untuk membangun model XGBoost setelah seleksi fitur.

3.4. Optimasi Hyperparameter

Optimasi *hyperparameter* dilakukan menggunakan GridSearchCV pada model XGBoost dengan 11 fitur hasil Boruta. Ruang pencarian terdiri atas $n_estimators$ {100, 200, 300}, max_depth {3, 5, 7}, $learning_rate$ {0,01; 0,05; 0,1}, $subsample$ {0,8; 1,0}, dan $colsample_bytree$ {0,8; 1,0}, sehingga terdapat 108 kombinasi. Setiap kombinasi dievaluasi dengan *Stratified 5-Fold Cross-Validation* ($shuffle=True$, $random_state=42$) dan F1-score sebagai metrik, sehingga proses GridSearchCV menghasilkan 540 proses *fitting*. Konfigurasi terbaik digunakan untuk membangun model akhir Boruta + GridSearchCV + XGBoost.

3.5. Evaluasi

Tiga konfigurasi dibandingkan, yaitu (1) XGBoost *baseline* dengan seluruh 36 fitur dan parameter bawaan tanpa seleksi fitur maupun optimasi; (2) Boruta + XGBoost dengan 11 fitur dan parameter bawaan; serta (3) Boruta + GridSearchCV + XGBoost dengan 11 fitur dan hasil optimasi. Ketiganya dievaluasi pada data uji menggunakan *accuracy*, *precision*, *recall*, F1-score, dan ROC-AUC, yang terakhir untuk melihat kemampuan model membedakan dua kelas pada berbagai nilai ambang keputusan [20]. Signifikansi perbedaan prediksi antar-model diuji dengan McNemar's test pada data uji yang sama dengan taraf signifikansi $\alpha = 0,05$. Efisiensi komputasi dinilai dari jumlah fitur serta waktu *fitting* dan prediksi; waktu *fitting* diukur sebagai median dari 10 pengulangan dan waktu prediksi sebagai median dari 30 pengulangan, masing-masing setelah satu kali pemanasan (*warm-up*) untuk mengurangi pengaruh inisialisasi.

3.6. Interpretasi SHAP

Model akhir (Boruta + GridSearchCV + XGBoost) diinterpretasikan menggunakan SHAP *TreeExplainer*, yang dirancang untuk model berbasis pohon dan dapat menjelaskan kontribusi fitur baik pada tingkat prediksi individual maupun struktur model secara global. Kepentingan fitur dihitung dari *mean absolute SHAP value* untuk menyusun peringkat fitur, sedangkan SHAP *summary plot* digunakan untuk melihat arah kontribusi fitur, yaitu nilai SHAP positif mendorong prediksi ke arah *Dropout* (label 1) dan nilai negatif mendorong prediksi ke arah *Graduate* (label 0).

3.7. Skenario Ketersediaan Fitur

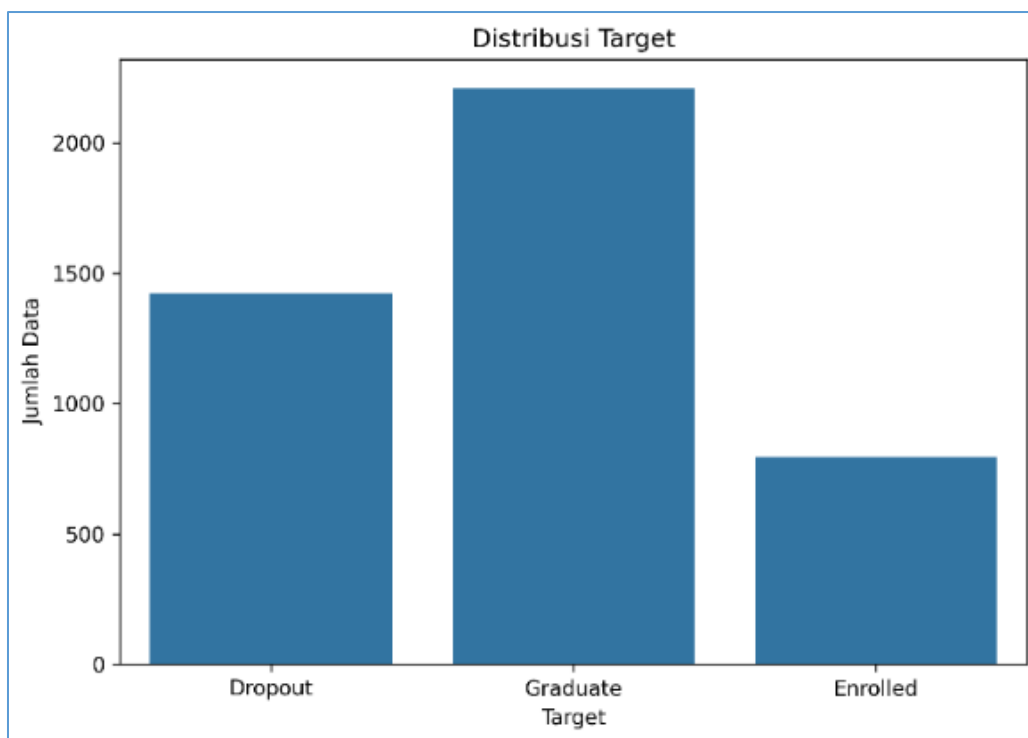
Analisis tambahan dilakukan untuk mengetahui kemampuan model ketika informasi akademik belum lengkap, sehingga relevan dengan penggunaan model pada tahap yang lebih awal. Tiga skenario dibandingkan, yaitu seluruh fitur (36 fitur), tanpa fitur semester 2 (30 fitur), dan tanpa fitur akademik (24 fitur) yang mengeluarkan seluruh kolom *Curricular units*. Ketiga skenario dilatih menggunakan XGBoost dengan parameter bawaan, tanpa seleksi fitur maupun optimasi *hyperparameter*, dan dievaluasi pada data uji yang sama dengan evaluasi utama, sehingga perbedaan performa hanya berasal dari ketersediaan fitur.

4 Hasil dan Pembahasan

Bagian ini menyajikan hasil seleksi fitur, performa model, efisiensi komputasi, interpretasi SHAP, dan skenario ketersediaan fitur. Seluruh hasil diperoleh dari 726 data uji yang tidak digunakan dalam pelatihan model.

4.1. Distribusi Kelas dan Seleksi Fitur Boruta

Gambar 1 memperlihatkan distribusi ketiga kelas sebelum *Enrolled* dikeluarkan. Kelas *Graduate* paling banyak sehingga distribusi kelas cukup tidak seimbang.



Gambar 1 Distribusi target

Dari 36 fitur awal, Boruta mengonfirmasi 11 fitur dan tidak menyisakan fitur berstatus *tentative*. Sebanyak 25 fitur tidak digunakan pada tahap berikutnya, sehingga jumlah fitur berkurang 69,44%. Fitur terpilih ditunjukkan pada Tabel 2.

Tabel 2 Fitur terpilih hasil seleksi fitur boruta

No	Fitur Terpilih
1.	<i>Course</i>
2.	<i>Admission grade</i>
3.	<i>Tuition fees up to date</i>
4.	<i>Scholarship holder</i>
5.	<i>Age at enrollment</i>
6.	<i>Curricular units 1st sem (evaluations)</i>
7.	<i>Curricular units 1st sem (approved)</i>
8.	<i>Curricular units 1st sem (grade)</i>
9.	<i>Curricular units 2nd sem (evaluations)</i>
10.	<i>Curricular units 2nd sem (approved)</i>
11.	<i>Curricular units 2nd sem (grade)</i>

Boruta menghasilkan subset fitur yang lebih ringkas, tetapi fitur yang tidak terpilih tidak dapat langsung dianggap tidak memiliki informasi terhadap status mahasiswa karena hasil seleksi bergantung pada mekanisme dan konfigurasi Boruta.

4.2. Pemodelan dan Optimasi XGBoost

Model *baseline* (36 fitur) dan model Boruta + XGBoost (11 fitur) dievaluasi pada 726 data uji, dan hasilnya dibandingkan pada Subbab 4.3. Optimasi *hyperparameter* pada 11 fitur hasil Boruta, dengan ruang pencarian dan skema validasi pada Subbab 3.4, menghasilkan konfigurasi terbaik yang ditunjukkan pada Tabel 3 beserta rata-rata F1-score validasi silang.

Tabel 3 Hasil GridSearchCV pada model boruta dan XGBoost

Parameter	Ruang Pencarian	Nilai Terbaik
n_estimators	100, 200, 300	300
max_depth	3, 5, 7	3
learning_rate	0,01; 0,05; 0,1	0,05
subsample	0,8; 1,0	0,8
colsample_bytree	0,8; 1,0	1,0
Best CV F1-score	–	0,8696

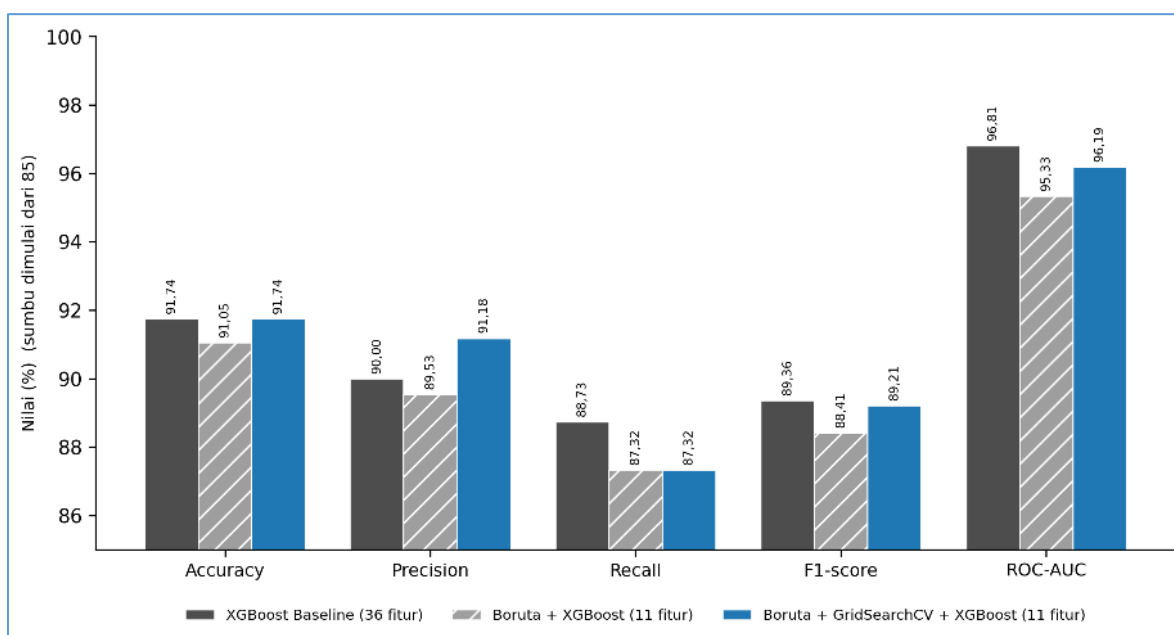
Konfigurasi tersebut digunakan untuk membangun model akhir Boruta + GridSearchCV + XGBoost. Nilai terbaik n_estimators (300) berada pada batas atas ruang pencarian, sehingga ruang yang lebih luas belum dieksplorasi.

4.3. Perbandingan Performa Model

Pengaruh seleksi fitur Boruta dan optimasi *hyperparameter* terhadap kinerja XGBoost ditunjukkan pada Tabel 4 dan Gambar 2.

Tabel 4 Perbandingan performa seluruh model

Model	Fitur	Accuracy	Precision	Recall	F1	ROC-AUC
XGBoost <i>Baseline</i>	36	91,74%	90,00%	88,73%	89,36%	96,81%
Boruta + XGBoost	11	91,05%	89,53%	87,32%	88,41%	95,33%
Boruta + GridSearchCV + XGBoost	11	91,74%	91,18%	87,32%	89,21%	96,19%



Gambar 2 Perbandingan metrik evaluasi tiga model (sumbu vertikal dimulai dari 85%)

Pengurangan fitur dari 36 menjadi 11 menurunkan seluruh metrik dibandingkan *baseline*, dengan penurunan terbesar pada ROC-AUC (1,48 poin persentase). GridSearchCV memulihkan *accuracy* ke 91,74% dan menaikkan *precision*, sedangkan *recall* dan ROC-AUC tetap sedikit di bawah *baseline* (Tabel 4). Dengan demikian, seleksi fitur dan optimasi tidak meningkatkan seluruh

metrik sekaligus. Model 11 fitur hasil optimasi hanya menyamai *accuracy baseline*, sehingga terdapat *trade-off* antara penyederhanaan fitur dan performa klasifikasi.

Selisih performa antar-model relatif kecil. *Accuracy* tertinggi dan terendah hanya berbeda 0,69 poin persentase, setara lima dari 726 data uji. McNemar's test menghasilkan nilai p sebesar 0,4990 (*baseline* dan Boruta + XGBoost), 0,8676 (*baseline* dan Boruta + GridSearchCV + XGBoost), serta 0,3827 (Boruta + XGBoost dan Boruta + GridSearchCV + XGBoost). Seluruh nilai p berada di atas $\alpha = 0,05$, sehingga tidak terdapat perbedaan signifikan secara statistik antar-model. Pemilihan model akhir dapat mempertimbangkan efisiensi komputasi (Subbab 4.4) dan kemudahan interpretasi (Subbab 4.5), dengan catatan bahwa temuan ini berasal dari satu kali pembagian data.

4.4. Efisiensi Komputasi

Waktu *fitting* dan prediksi, dengan prosedur pengukuran pada Subbab 3.5, ditunjukkan pada Tabel 5.

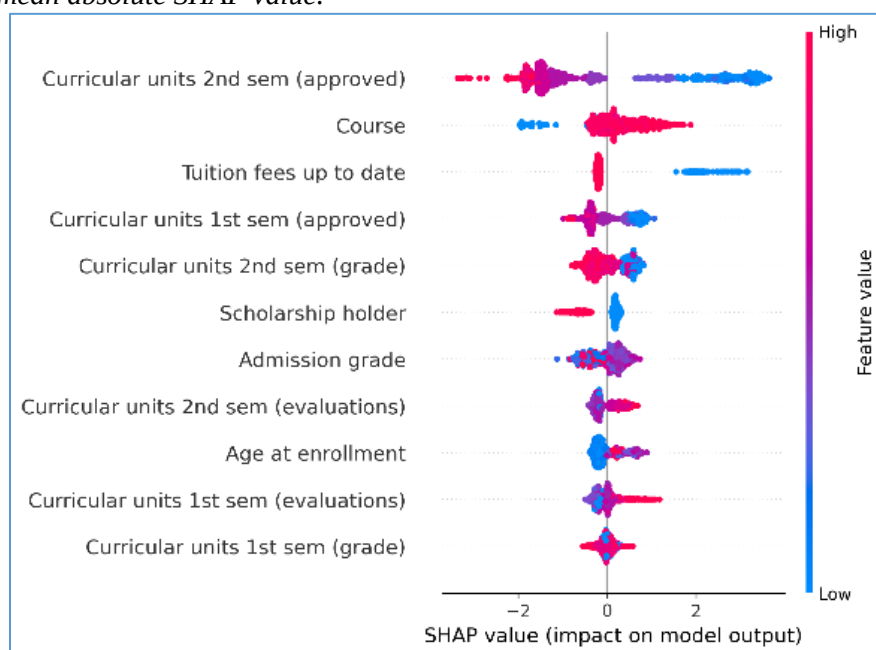
Tabel 5 Perbandingan efisiensi waktu *fitting* dan prediksi

Model	Fitur	<i>Fitting</i> (detik)	Prediksi (detik)
XGBoost <i>Baseline</i>	36	0,2050	0,0214
Boruta + XGBoost	11	0,1256	0,0115
Boruta + GridSearchCV + XGBoost	11	0,2256	0,0150

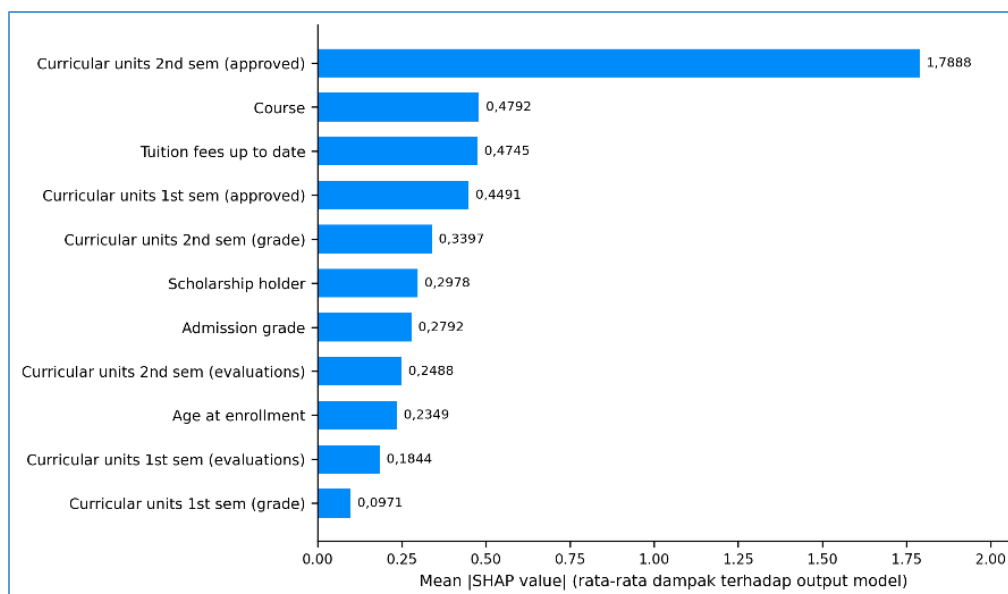
Reduksi fitur oleh Boruta (dari 36 menjadi 11 fitur) menurunkan waktu *fitting* sekitar 38,7% dan waktu prediksi sekitar 46,3% dibandingkan *baseline*. Setelah optimasi, waktu *fitting* justru naik sekitar 10% dari *baseline* karena konfigurasi terbaik menggunakan *n_estimators*=300, jauh lebih banyak daripada nilai bawaan (100), sedangkan waktu prediksi tetap lebih rendah sekitar 29,9%. Dengan demikian, reduksi fitur secara konsisten mempercepat prediksi, sedangkan waktu *fitting* bergantung pada kompleksitas model hasil optimasi. Waktu tersebut tidak mencakup biaya komputasi proses Boruta dan GridSearchCV itu sendiri.

4.5. Interpretasi Model dengan SHAP

Interpretasi dilakukan pada model akhir (Boruta + GridSearchCV + XGBoost). Gambar 3 menampilkan SHAP *summary plot*, dan Gambar 4 menampilkan peringkat kontribusi fitur berdasarkan *mean absolute SHAP value*.



Gambar 3 SHAP *summary plot* model boruta + GridSearchCV + XGBoost



Gambar 4 SHAP feature importance (mean absolute SHAP value)

Gambar 3 dan Gambar 4 memperlihatkan bahwa *Curricular units 2nd sem (approved)* memiliki kontribusi terbesar terhadap prediksi (*mean absolute SHAP* 1,7888), jauh di atas *Course* (0,4792) dan *Tuition fees up to date* (0,4745). Nilai rendah pada fitur pertama cenderung mendorong prediksi ke arah *Dropout*, sedangkan nilai tinggi mendorong ke arah *Graduate*. Status pembayaran tanpa tunggakan juga cenderung mendorong prediksi ke arah *Graduate*. Dengan demikian, jumlah mata kuliah yang disetujui dan kelancaran pembayaran menjadi sinyal utama yang dipakai model.

Terdapat tiga catatan dalam membaca hasil ini. Pertama, *Course* merupakan variabel nominal berkode numerik, sehingga warna pada Gambar 3 tidak dapat diartikan sebagai program studi berkode tinggi atau rendah. Kedua, kontribusi fitur menggambarkan keputusan model dan tidak dapat diartikan sebagai hubungan sebab-akibat terhadap *dropout* mahasiswa. Ketiga, fitur yang paling berpengaruh baru tersedia setelah dua semester berjalan, sehingga model dalam bentuknya saat ini lebih sesuai sebagai alat identifikasi risiko setelah data akademik tersebut tersedia (Subbab 4.6). Berbeda dengan Carballo-Mendivil dkk. [5] yang menggunakan data pra-pendaftaran, model ini tidak dirancang sebagai sistem peringatan dini sejak pendaftaran, melainkan untuk menganalisis perubahan performa seiring bertambahnya informasi akademik.

4.6. Analisis Skenario Ketersediaan Fitur

XGBoost dilatih pada tiga skenario ketersediaan fitur (Subbab 3.7), dan hasilnya ditunjukkan pada Tabel 6.

Tabel 6 Performa model pada skenario ketersediaan fitur

Skenario	Fitur	Accuracy	Precision	Recall	F1	ROC-AUC
Semua fitur	36	91,74%	90,00%	88,73%	89,36%	96,81%
Tanpa fitur semester 2	30	90,22%	87,63%	87,32%	87,48%	94,59%
Tanpa fitur akademik (<i>Curricular units</i>)	24	74,93%	69,32%	64,44%	66,79%	81,28%

Tanpa fitur semester 2, *accuracy* turun 1,52 poin persentase dari model dengan seluruh fitur, sehingga informasi semester pertama sudah memberikan performa yang cukup baik dan membuka peluang penggunaan model pada akhir semester pertama. Tanpa seluruh fitur akademik, *accuracy* turun tajam sekitar 16,8 poin persentase dari model dengan seluruh fitur. *Recall* yang rendah berarti

sekitar 36% mahasiswa *Dropout* tidak terdeteksi, sehingga model kurang sesuai digunakan saat pendaftaran.

5 Kesimpulan

Evaluasi pengaruh seleksi fitur Boruta dan optimasi *hyperparameter* terhadap performa XGBoost dalam memprediksi status *Graduate* dan *Dropout* memberikan temuan berikut. Boruta mengurangi jumlah fitur secara substansial, dan model Boruta + GridSearchCV + XGBoost mempertahankan *accuracy* yang sama dengan *baseline* meskipun tidak unggul pada seluruh metrik, dengan McNemar's test yang tidak menemukan perbedaan signifikan antar-model. Pengurangan fitur menurunkan waktu prediksi pada kedua model 11 fitur, sedangkan waktu *fitting* lebih rendah hanya pada model tanpa optimasi karena konfigurasi hasil GridSearchCV menggunakan lebih banyak estimator. SHAP mengidentifikasi *Curricular units 2nd sem (approved)*, *Course*, dan *Tuition fees up to date* sebagai fitur dengan kontribusi terbesar, dan analisis skenario menunjukkan bahwa informasi akademik berperan besar dalam prediksi. Meskipun demikian, temuan ini terbatas pada satu dataset UCI, klasifikasi dua kelas, dan satu kali pembagian data sehingga stabilitas hasil belum diuji, serta fitur paling berpengaruh baru tersedia setelah semester kedua. Penelitian selanjutnya disarankan menerapkan *repeated stratified cross-validation*, menguji model pada dataset dari institusi atau periode lain, dan mengevaluasi model pada akhir semester pertama untuk menilai kemampuan prediksi pada tahap yang lebih awal.

Referensi

- [1] A. J. Fernández-García, J. C. Preciado, F. J. Melchor-González, R. Rodríguez-Echeverría, J. M. Conejero, and F. Sánchez-Figueroa, "A Real-Life Machine Learning Experience for Predicting University Dropout at Different Stages using Academic Data," *IEEE Access*, Vol. 9, pp. 133076–133090, 2021, DOI: 10.1109/ACCESS.2021.3115851.
- [2] L. G. R. Putra, D. D. Prasetya, and Mayadi, "Student Dropout Prediction using Random Forest and XGBoost Method," *INTENSIF*, Vol. 9, No. 1, pp. 147–157, 2025, DOI: 10.29407/intensif.v9i1.21191.
- [3] V. Realinho, J. Machado, L. Baptista, and M. V. Martins, "Predicting Student Dropout and Academic Success," *Data*, Vol. 7, No. 11, p. 146, 2022, DOI: 10.3390/data7110146.
- [4] M. L. Damar Sakti et al., "The Effect of Sample Size on the Stability of XGBoost Model Performance in Predicting Student Study Period," *BAREKENG J. Ilmu Mat. dan Terap.*, Vol. 19, No. 4, pp. 2679–2692, 2025, DOI: 10.30598/barekengvol19iss4pp2679-2692.
- [5] B. Carballo-Mendivil, A. Arellano-González, N. J. Ríos-Vázquez, and M. del P. Lizardi-Duarte, "Predicting Student Dropout from Day One: XGBoost-based Early Warning System using Pre-Enrollment Data," *Appl. SCI.*, Vol. 15, No. 16, p. 9202, 2025, DOI: 10.3390/app15169202.
- [6] S. Praptomo, A. Risman, and R. M. Suri, "Explainable Machine Learning for Predicting Student Dropout and Academic Success using XGBoost and SHAP," *J. Electr. Eng. Comput.*, Vol. 8, No. 1, pp. 222–231, 2026, DOI: 10.33650/jeeecom.v8i1.16978.
- [7] H. Ahmadpour, O. Bazrafshan, E. Rafiei-Sardooi, H. Zamani, and T. Panagopoulos, "Gully Erosion Susceptibility Assessment in the Kondoran Watershed using Machine Learning Algorithms and the Boruta Feature Selection," *Sustainability*, Vol. 13, No. 18, p. 10110, 2021, DOI: 10.3390/su131810110.
- [8] A. Mincevičius, "Interpretable Machine Learning Approaches for Student Dropout Prediction in Higher Education," *New Trends Comput. SCI.*, Vol. 4, No. 1, pp. 146–160, 2026, DOI: 10.3846/ntcs.2026.27184.
- [9] E. Elgeldawi, A. Sayed, A. R. Galal, and A. M. Zaki, "Hyperparameter Tuning for Machine Learning Algorithms used for Arabic Sentiment Analysis," *Informatics*, Vol. 8, No. 4, p. 79, 2021, DOI: 10.3390/informatics8040079.
- [10] M. Li, H. Sun, Y. Huang, and H. Chen, "Shapley Value: from Cooperative Game to Explainable Artificial Intelligence," *Auton. Intell. Syst.*, Vol. 4, No. 1, p. 2, 2024, DOI:

- 10.1007/s43684-023-00060-8.
- [11] R. Mitchell, E. Frank, and G. Holmes, “GPUShap: Massively Parallel Exact Calculation of SHAP Scores for Tree Ensembles,” *PeerJ Comput. SCI.*, Vol. 8, p. e880, 2022, DOI: 10.7717/peerj-cs.880.
 - [12] I. K. Nti and S. Ramanayake, “Explainable Machine Learning for Student Dropout Prediction and Tailored Interventions in Online Personalized Education,” *Discov. Artif. Intell.*, Vol. 6, No. 1, p. 288, 2026, DOI: 10.1007/s44163-026-01016-6.
 - [13] M. Quimiz-Moreira, R. Delgadillo, J. Parraga-Alava, N. Maculan, and D. Mauricio, “Factors, Prediction, Explainability, and Simulating University Dropout through Machine Learning: A Systematic Review, 2012–2024,” *Computation*, Vol. 13, No. 8, p. 198, 2025, DOI: 10.3390/computation13080198.
 - [14] R. Goran et al., “Identifying and Understanding Student Dropouts using Metaheuristic Optimized Classifiers and Explainable Artificial Intelligence Techniques,” *IEEE Access*, Vol. 12, pp. 122377–122400, 2024, DOI: 10.1109/ACCESS.2024.3446653.
 - [15] A. M. Priyatno, Y. Ningsih, R. J. Farhas, F. I. Firmananda, R. K. Sari, and Aryadi, “Multi-Stage Feature Selection for Optimizing Student Dropout Prediction,” *J. Eng. Technol. Ind. Appl.*, Vol. 11, No. 55, pp. 86–95, 2025, DOI: 10.5935/jetia.v11i55.1910.
 - [16] E. Alzabidi, Ö. Yakut, S. Saad, and O. Findik, “SOD-FE: A Supervised Outlier Detection and Feature Engineering Approach for Student Dropout Prediction,” *SCI. Rep.*, Vol. 16, No. 1, p. 25906, 2026, DOI: 10.1038/s41598-026-56591-6.
 - [17] F. Y. Dewi, B. R. Pratama, and S. Sunaryono, “Perbandingan Grid Search dan Random Search untuk Optimasi Hyperparameter Random Forest pada Klasifikasi Kanker Payudara,” *Bull. Comput. SCI. Res.*, Vol. 6, No. 4, pp. 1489–1497, 2026, DOI: 10.47065/bulletincsr.v6i4.1087.
 - [18] S. Kapoor and A. Narayanan, “Leakage and the Reproducibility Crisis in Machine-Learning-based Science,” *Patterns*, Vol. 4, No. 9, p. 100804, 2023, DOI: 10.1016/j.patter.2023.100804.
 - [19] S. Bag, K. Gupta, and S. Deb, “A Review and Recommendations on Variable Selection Methods in Regression Models for Binary Data,” *arXiv Prepr. arXiv2201.06063*, 2022, DOI: 10.48550/arXiv.2201.06063.
 - [20] G. Canbek, T. Taskaya Temizel, and S. Sagiroglu, “PToPI: A Comprehensive Review, Analysis, and Knowledge Representation of Binary Classification Performance Measures/Metrics,” *SN Comput. SCI.*, Vol. 4, p. 13, 2023, DOI: 10.1007/s42979-022-01409-1.