JRMM – Jurnal Riset Mahasiswa Matematika, Volume 6 (2) (2026), Pages 33-44

Research Article

Implementasi Metode MOORA Berbobot untuk Pemeringkatan Model *Machine Learning* pada Deteksi Mahasiswa Berisiko *Dropout*

Izzar Suly Nashrudin*, Elif Thoyyibah Rahmawati, Nur Intan Aprilia, Siti Syarifah Jamalia, Dzikry Fikrillah, and Risma Muslimah

Universitas Islam Negeri Maulana Malik Ibrahim Malang, Malang, Indonesia

Article History

Received 08 Juni 2026
Revised 25 Oktober 2026
Accepted 31 Oktober 2026
Published 30 Desember 2026

Check for updates

Copyright © 2026 by Authors, Published by JRMM Group. This is an open access article under the CC BY-SA License.

Abstract. The student *dropout* rate in higher education is a real challenge that indirectly threatens institutional reputation and students' own futures. Prediction using *machine learning* has developed rapidly, but model evaluation sometimes relies on accuracy metrics without considering computational resources such as computational time and model size. This research aims to design a multicriteria decision-making approach using *Multi-Objective Optimization on the Basis of Ratio Analysis* (`MOORA`) to rank *dropout* prediction models, including *Logistic Regression*, *Gaussian Naive Bayes*, `KNN`, *Decision Tree*, *Random Forest*, `SVM`, `LDA`, and `QDA`. The study was conducted on the *Kaggle* platform using a *CPU Multi-Thread* configuration with four *seeds* and ten repetitions per *seed*. The evaluation used benefit criteria consisting of accuracy, precision, *recall*, F1-*score*, and `ROC-AUC`, as well as cost criteria consisting of *log loss*, computational time, model size, `FPR`, and `FNR`. The results show that *Logistic Regression* ranked first with $Y_1 = 0.170760$, followed by `LDA` with $Y_7 = 0.167518$ and `QDA` with $Y_8 = 0.109363$. *Random Forest* dropped to seventh place due to high computational cost and model size. In the inference scenario involving 794 *Enrolled* students, the top three models predicted 30.23%–53.65% of students as being at risk of *dropout*. This research is designed to support operationally feasible and resource-efficient model selection for academic early warning systems.

Keywords: Decision; Deep Learning; Dropout; Machine Learning; MOORA; Student

Abstrak. Tingkat *dropout* pada mahasiswa di perguruan tinggi menjadi tantangan nyata yang tidak langsung mengancam reputasi institusi dan masa depan mahasiswa itu sendiri. Prediksi menggunakan *machine learning* telah berkembang pesat, tetapi evaluasi model terkadang bergantung pada metrik akurasi dan tanpa mempertimbangkan sumber daya seperti waktu komputasi dan ukuran model. Tujuan riset untuk merancang pengambilan keputusan multikriteria menggunakan *Multi-Objective Optimization on the Basis of Ratio Analysis* (`MOORA`) untuk memeringkat model prediksi *dropout* seperti *Logistic Regression*, *Gaussian Naive Bayes*, `KNN`, *Decision Tree*, *Random Forest*, `SVM`, `LDA`, dan `QDA`. Penelitian menggunakan *platform Kaggle* berbasis *CPU Multi-Thread* dengan 4 *seed* dan 10 kali perulangan per *seed*. Dengan kriteria keuntungan (akurasi, presisi, *recall*, F1-*score*, dan `ROC-AUC`) dan kriteria biaya (*log loss*, waktu komputasi, ukuran model, `FPR`, dan `FNR`). Hasil riset menunjukkan *Logistic Regression* menduduki peringkat pertama dengan $Y_1 = 0{,}170760$, diikuti `LDA` dengan $Y_7 = 0{,}167518$ dan `QDA` dengan $Y_8 = 0{,}109363$. *Random Forest* turun ke peringkat ketujuh akibat biaya komputasi dan ukuran model tinggi. Pada skenario inferensi 794 mahasiswa *Enrolled*, tiga model teratas memprediksi 30,23%–53,65% mahasiswa berisiko *dropout*. Penelitian ini dirancang untuk seleksi model yang layak operasional dan hemat sumber daya untuk sistem peringatan dini akademik.

Kata kunci: *Dropout*; Keputusan; *Machine Learning*; MCDM; Mahasiswa; MOORA

1. Pendahuluan

Tingkat *dropout* mahasiswa merupakan isu krusial dalam pendidikan tinggi global karena berdampak buruk pada stabilitas finansial institusi, efisiensi alokasi sumber daya, dan masa depan sosioekonomi mahasiswa [1], [2]. Bagi mahasiswa, fenomena ini memicu kerugian akademik dan finansial, sementara bagi universitas dapat menurunkan tingkat kelulusan serta merusak reputasi lembaga [1]. Oleh karena itu, diperlukan strategi proaktif melalui *Educational Data Mining* (`EDM`) dan *Learning Analytics* untuk mengeksplorasi data akademis, demografis, dan sosioekonomi guna membangun sistem peringatan dini (*early warning system*) yang mampu mengidentifikasi mahasiswa berisiko sebelum mengalami kegagalan studi [3], [4].

Berbagai penelitian terdahulu telah memanfaatkan algoritma *machine learning* seperti *Decision Tree*, *Random Forest*, dan *Support Vector Machine* (`SVM`) untuk memprediksi risiko *dropout* [5]. Namun, mayoritas evaluasi model masih bergantung pada metrik tunggal seperti akurasi, yang kurang memadai akibat bias terhadap kelas mayoritas pada karakteristik data pendidikan yang tidak seimbang (*imbalanced data*) [5]. Di sisi lain, implementasi operasional di perguruan tinggi juga menuntut pertimbangan kendala biaya (*cost criteria*) seperti waktu komputasi yang lama atau ukuran penyimpanan model yang besar [6], [7]. Karena adanya pertentangan (*trade-off*) antarberbagai indikator ini, pemilihan model prediksi *dropout* optimal pada dasarnya merupakan masalah pengambilan keputusan multikriteria (*Multi-Criteria Decision Making*/`MCDM`) [6], [7].

Penelitian ini menerapkan metode `MOORA` dengan pembobotan manual (*non-uniform weighting*) untuk mengevaluasi delapan arsitektur model klasifikasi secara ketat, yaitu *Logistic Regression*, *Gaussian Naive Bayes*, *K-Nearest Neighbors* (`KNN`), *Decision Tree*, *Random Forest*, `SVM`, `LDA`, dan `QDA` [8], [9]. Metode `MCDM` ini dipilih karena mampu mengintegrasikan

Corresponding author's email: izzarsuly33@gmail.com
DOI: https://doi.org/10.18860/JRMM.v6i2.43376 / p-ISSN: 2808-1552 | e-ISSN: 2808-4926 (BRIN | ISSN Portal)

kriteria keuntungan (*benefit*) dan biaya (*cost*) secara simultan lewat tahapan matematis yang ringkas dan stabil [10]. Evaluasi model dilakukan menggunakan sepuluh kriteria komprehensif, mencakup lima kriteria keuntungan (akurasi, presisi, *recall*, F1-*score*, `ROC-AUC`) dan lima kriteria biaya (*log loss*, waktu komputasi, ukuran model, `FPR`, `FNR`), dengan prioritas bobot lebih tinggi pada aspek penekanan *false negative* [8], [9]. Untuk menjamin stabilitas hasil, seluruh model dilatih menggunakan konfigurasi *CPU Multi-Thread* pada platform *Kaggle* melalui skenario 4 *random seed* dan 10 perulangan (total 40 kali pengujian per model) [8], [9]. Terakhir, seluruh delapan model dilatih ulang dan diterapkan pada skenario inferensi 794 mahasiswa aktif berstatus *Enrolled* untuk menghasilkan estimasi probabilitas risiko *dropout*, dengan tiga model teratas `MOORA` sebagai fokus analisis utama [8], [9].

Kebaruan (*novelty*) penelitian ini terletak pada integrasi pendekatan `EDM` dengan sistem pendukung keputusan `MOORA` untuk menilai kelayakan model secara holistik. Berbeda dengan literatur sebelumnya yang terisolasi pada metrik klasifikasi umum, studi ini secara eksplisit menguji keseimbangan antara kualitas probabilitas (*log loss*), ketepatan penargetan, dan efisiensi sumber daya komputasi. Hasil studi ini diproyeksikan dapat memberikan rekomendasi pemilihan model yang objektif serta menjadi pedoman praktis bagi perguruan tinggi dalam mengembangkan sistem peringatan dini yang hemat sumber daya namun berakurasi tinggi.

2. Metode

Bagian ini menguraikan dua tahap utama penelitian. Tahap pertama adalah desain eksperimen yang mencakup penggunaan *dataset*, prapemrosesan, pelatihan dan pengujian model, serta metrik evaluasi. Tahap kedua adalah implementasi metode `MOORA` untuk memeringkat model berdasarkan hasil eksperimen tersebut.

2.1. Desain eksperimen

Penelitian ini bersifat eksperimen komputasional yang berjalan di lingkungan *Kaggle* dengan semua konfigurasi alternatif model yang dijelaskan pada bagian Alternatif model menggunakan *CPU Multi-Thread*. Akselerasi `GPU` dinonaktifkan melalui `CUDA_VISIBLE_DEVICES=""`. Pada eksekusinya, jumlah *thread* yang diekspos ke pustaka `BLAS/OpenMP` adalah 4. Estimator yang mendukung paralelisme dikonfigurasi dengan `n_jobs=-1`, sedangkan *cross-validation* dijalankan secara berurutan melalui `n_jobs=1` untuk menghindari *nested oversubscription*. Pengaturan ini dipakai agar perbedaan waktu komputasi antarmodel tetap menggunakan `CPU` tanpa memanfaatkan akselerasi `GPU`.

Runtime berjalan pada *Python* 3.12.12, sedangkan versi paketnya ditunjukkan pada Tabel 1. Informasi ini penting karena perilaku estimator, parameter, dan implementasi utilitas seperti `OneHotEncoder`, `Pipeline`, `StratifiedKFold`, serta metrik evaluasi dapat berubah antarversi paket.

Tabel 1: Versi lingkungan dan paket *Python* yang digunakan pada eksperimen.

Komponen	Nama impor	Versi
Python	`python`	3.12.12
NumPy	`numpy`	2.0.2
Pandas	`pandas`	2.3.3
Matplotlib	`matplotlib`	3.10.0
Scikit-learn	`sklearn`	1.6.1
`ucimlrepo`	`ucimlrepo`	0.0.7
IPython	`IPython`	7.34.0

2.1.1. Penggunaan dataset

Dataset penelitian ini adalah *Predict Students' Dropout and Academic Success* dari *UCI Machine Learning Repository* (`ID` 697) [11]. *Dataset* ini berisi 4.424 mahasiswa dan 36 fitur yang menjelaskan masing-masing kondisi sosial dan ekonomi serta kolom target yang menunjukkan tiga status akademik, yaitu *Graduate*, *Dropout*, dan *Enrolled*. Distribusi kelas pada *dataset* asal terdiri atas 2.209 mahasiswa *Graduate*, 1.421 mahasiswa *Dropout*, dan 794 mahasiswa *Enrolled*.

Dataset ini juga punya karakteristik yang unik, seperti distribusi kelas yang tidak seimbang (*imbalance data*), data sudah melalui proses ketat dalam penanganan *outlier*, dan tidak ada nilai yang hilang (*No Missing Values*). Dalam klasifikasi *dropout*, ketidakseimbangan data dapat menyebabkan metrik akurasi menjadi condong pada satu kelas mayoritas dibandingkan mengenali kelas minoritas [12]. Karena penelitian ini berfokus pada deteksi risiko *dropout*, maka pada eksperimen utama dirumuskan sebagai klasifikasi biner *Graduate* versus *Dropout*. Kelas *Dropout* didefinisikan sebagai kelas positif $y = 1$, sedangkan kelas *Graduate* didefinisikan sebagai kelas negatif $y = 0$.

Kelas dengan status *Enrolled* tidak digunakan dalam pelatihan maupun evaluasi utama, tetapi digunakan pada skenario kedua, yaitu skenario inferensi sesuai dengan sistem peringatan dini akademik. Dalam kehidupan nyata, mahasiswa aktif belum memiliki status akhir, sehingga model tidak dipakai untuk memprediksi label *Graduate* atau *Dropout*, melainkan untuk menghasilkan estimasi risiko yang dapat membantu dalam menentukan keputusan lebih lanjut [13]. Oleh karena itu, pemisahan antara data *Graduate* dan *Dropout* untuk evaluasi dan data *Enrolled* untuk inferensi risiko merupakan tujuan operasional prediksi *dropout*.

Secara formal, *dataset* terlabel untuk eksperimen utama dinyatakan sebagai

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}, \qquad \mathbf{x}_i \in \mathbb{R}^d, \quad y_i \in \{0, 1\}, \tag{1}$$

dengan definisi label

$$y_i = \begin{cases} 1, & \text{jika status akhir mahasiswa adalah } \textit{Dropout}, \\ 0, & \text{jika status akhir mahasiswa adalah } \textit{Graduate}. \end{cases} \tag{2}$$

Setelah data *Enrolled* dikeluarkan dari klasifikasi biner, jumlah data yang digunakan dalam eksperimen utama menjadi $n = 3.630$, terdiri atas 2.209 mahasiswa *Graduate* dan 1.421 mahasiswa *Dropout*, sebagaimana dinyatakan pada (1) dan (2).

2.1.2. Tahap preprocessing

Tahap prapemrosesan dilakukan menggunakan *pipeline* yang membuat semua transformasi data dipelajari dari data latih pada setiap pembagian data. Setelah data dengan target

Enrolled dipisahkan, fitur masukan dipecah menjadi fitur numerik dan fitur kategorikal berdasarkan tipe datanya. Fitur numerik diproses melalui imputasi median, kemudian distandardisasi menggunakan *z-score*. Standardisasi fitur numerik x_{ij} pada (3) didefinisikan sebagai

$$z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}, \tag{3}$$

dengan μ_j dan σ_j masing-masing merupakan rata-rata dan simpangan baku fitur ke-j. Sedangkan, fitur kategorikal diproses melalui imputasi modus, kemudian dikonversi menjadi numerik menggunakan *one-hot encoding*. *Encoder* dikonfigurasi untuk mengabaikan kategori yang tidak muncul pada data latih tetapi muncul pada data uji, sehingga proses tersebut mencegah *data leakage*.

2.1.3. Tahap pelatihan dan pengujian

Tahap selanjutnya merupakan **eksperimen utama** menggunakan rasio data latih dan data uji sebesar 80:20 dengan stratifikasi label. Stratifikasi dilakukan karena proporsi kelas *Graduate* dan *Dropout* tidak seimbang yang menyebabkan pembagian data acak dapat menghasilkan *subset* uji yang tidak mendekati distribusi kelas sebenarnya [12]. Penggunaan *random seed* di mana $s \in S$, dengan

$$S = \{0, 7, 42, 43\}, \tag{4}$$

digunakan untuk pembentukan pasangan data latih dan data uji

$$(\mathcal{D}^{(s)}_{\text{train}}, \mathcal{D}^{(s)}_{\text{test}}) = \text{Split}_{0.8:0.2}(\mathcal{D}; s), \tag{5}$$

dengan syarat distribusi kelas pada $\mathcal{D}^{(s)}_{\text{train}}$ dan $\mathcal{D}^{(s)}_{\text{test}}$ pada (5) tetap melalui stratifikasi label.

Penggunaan beberapa *random seed* pada (4) diperlukan karena hasil evaluasi model di akhir dapat dipengaruhi oleh pembagian data latih dan data uji yang berbeda. Jika evaluasi model hanya bergantung pada satu *seed* saja, maka hasil yang diperoleh terlalu condong pada satu konfigurasi partisi data. Kondisi ini menyebabkan hasil pemeringkatan model menjadi bias terhadap satu *split* tertentu. Oleh karena itu, evaluasi harus dilakukan pada 4 *seed* yang berbeda-beda untuk mengukur seberapa stabil performa model dengan variasi partisi data.

Seed hanya sebagai pengontrol sumber keacakan pada pembagian data, *K-Fold Cross Validation*, dan komponen stokastik pada estimator yang menggunakan `random_state`. Untuk setiap *seed*, proses diulang sebanyak sepuluh kali, sehingga model dievaluasi pada 40 *run*, sebagaimana berikut

$$N_{\text{run}} = |S| \times R = 4 \times 10 = 40, \tag{6}$$

dengan $R = 10$ menyatakan jumlah perulangan pada setiap *seed*. Setiap hasil evaluasi model ke-k pada *seed* ke-s dan pengulangan ke-r menghasilkan nilai metrik

$$\mathbf{m}_k^{(s,r)} = \big(\text{Accuracy}, \text{Precision}, \text{Recall}, \text{F1}, \text{ROC-AUC}, \text{LogLoss}, \text{Time}, \text{Size}, \text{FPR}, \text{FNR}\big)_k^{(s,r)}. \tag{7}$$

Nilai akhir setiap metrik yang digunakan dalam matriks keputusan diperoleh dari rata-rata seluruh pengukuran:

$$\bar{\mathbf{m}}_k = \frac{1}{|S|R} \sum_{s \in S} \sum_{r=1}^{R} \mathbf{m}_k^{(s,r)}. \tag{8}$$

Dengan cara ini, pemeringkatan model tidak ditentukan oleh satu hasil percobaan tunggal, melainkan performa rata-rata model pada beberapa konfigurasi acak.

Pada setiap percobaan, untuk melihat konsistensi model dilakukan validasi silang seperti pada *subset* data latih yang memakai *stratified k-fold cross-validation* dengan lima *fold*. Untuk setiap *seed* s, *fold* validasi dibentuk sebagai

$$\mathcal{D}^{(s)}_{\text{train}} = \bigcup_{\ell=1}^{5} \mathcal{F}^{(s)}_{\ell}, \tag{9}$$

masing-masing *fold* $\mathcal{F}^{(s)}_{\ell}$ pada (9) tetap mengikuti proporsi kelas melalui stratifikasi. Sedangkan evaluasi utama tetap pada data uji *hold-out*. Oleh karena itu, nilai masukan pada matriks keputusan *Multi-Objective Optimization by Ratio Analysis* (`MOORA`) adalah rata-rata metrik pada data uji *hold-out*, bukan nilai *cross-validation*.

Skenario kedua alurnya hampir sama dengan skenario pertama, yaitu melatih model pada klasifikasi biner *Graduate* versus *Dropout*, lalu menerapkannya pada seluruh data 794 mahasiswa berstatus *Enrolled*. Pada skenario ini, *random state* dipilih hanya 42. Karena status akhir mahasiswa *Enrolled* belum diketahui, maka tidak digunakan untuk menghitung akurasi, presisi, *recall*, atau metrik klasifikasi berbasis label aktual. Tujuan skenario ini hanya sebagai simulasi inferensi risiko, dengan keluaran berupa label prediksi, probabilitas risiko *dropout* $\hat{p}_i$, dan jumlah mahasiswa aktif yang diklasifikasikan sebagai berisiko *dropout* berfungsi sebagai pendukung keputusan lebih lanjut [13].

2.1.4. Metrik Evaluasi

Evaluasi menggunakan *confusion matrix* pada (10) untuk kelas positif *Dropout*:

$$\mathbf{C} = \begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}. \tag{10}$$

Metrik klasifikasi pada (11)–(16) didefinisikan sebagai

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{11}$$

$$Precision = \frac{TP}{TP + FP}, \tag{12}$$

$$Recall = \frac{TP}{TP + FN}, \tag{13}$$

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}, \tag{14}$$

$$\texttt{FPR} = \frac{FP}{FP + TN}, \tag{15}$$

$$\texttt{FNR} = \frac{FN}{FN + TP}. \tag{16}$$

Log loss pada (17) digunakan untuk menilai kualitas probabilitas:

$$\mathcal{L}_{\log} = -\frac{1}{n} \sum_{i=1}^{n} \left[y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i) \right]. \tag{17}$$

ROC-AUC digunakan untuk menilai kemampuan pemeringkatan probabilitas terhadap pasangan sampel positif-negatif, di mana model memberi skor lebih tinggi kepada sampel *Dropout* daripada sampel *Graduate*. Di sisi lain, metrik *recall* dan FNR perlu diperhatikan karena *false negative* berarti mahasiswa yang benar-benar berisiko *Dropout* tidak terdeteksi, sedangkan FPR menunjukkan *false positive*, yaitu mahasiswa yang tidak berisiko tetapi salah diprediksi berisiko.

2.2. Implementasi *Multi-Objective Optimization by Ratio Analysis* (MOORA)

2.2.1. Alternatif model

Himpunan alternatif keputusan didefinisikan sebagai berikut

$$\mathcal{M} = \{M_1, M_2, \ldots, M_8\}. \tag{18}$$

Model dalam $\mathcal{M}$ pada (18) yang akan dievaluasi meliputi *Logistic Regression*, *Gaussian Naive Bayes*, *K-Nearest Neighbors* (KNN), *Decision Tree*, *Random Forest*, *Support Vector Machine* (SVM), *Linear Discriminant Analysis* (LDA), dan *Quadratic Discriminant Analysis* (QDA). Pemilihan alternatif model ini bertujuan mencakup beberapa model *machine learning* dengan algoritma yang berbeda, seperti model *linear*, model probabilistik, model berbasis jarak, model berbasis *tree*, model *margin non-linear*, dan model diskriminan statistik.

Model-model tersebut telah disiapkan beberapa konfigurasi per model seperti pada Tabel 2 yang cocok untuk menghadapi karakteristik data yang digunakan pada penelitian ini yang sebelumnya telah dijelaskan pada bagian Penggunaan *dataset* serta penggunaan CPU sebagaimana dijelaskan di awal pada bagian Desain eksperimen.

Pemilihan model sudah sejalan dengan studi prediksi *dropout* dan performa akademik yang banyak membandingkan algoritma klasik pada data pendidikan [14], [15], [16].

2.2.2. Penyusunan matriks keputusan

Model yang sudah dievaluasi menggunakan prosedur eksperimen berulang yang dijelaskan pada bagian Tahap pelatihan dan pengujian, hasil pengukuran disusun menjadi matriks keputusan sebagai masukan metode MOORA. Matriks keputusan berukuran $m \times q$ disusun dengan $m = 8$ baris yang mewakili alternatif model dan $q = 10$ kolom sesuai dengan (7) yang mewakili kriteria evaluasi:

$$\mathbf{X} = [x_{ij}]_{m\times q}, \qquad i = 1, \ldots, m, \quad j = 1, \ldots, q. \tag{19}$$

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1q} \\ x_{21} & x_{22} & \cdots & x_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mq} \end{bmatrix} \tag{20}$$

Setiap elemen x_{ij} menyatakan nilai rata-rata metrik ke-j untuk model ke-i yang dihitung dari 40 *run* ulangan sesuai dengan (6) dan (8).

Lima kolom pertama ($j = 1, \ldots, 5$) merupakan kriteria keuntungan atau *benefit*, yaitu akurasi, presisi, *recall*, F1-*score*, dan ROC-AUC, yang bernilai lebih tinggi untuk kinerja yang lebih bagus. Lima kolom berikutnya ($j = 6, \ldots, 10$) adalah kriteria biaya atau *cost*, yaitu *log loss*, waktu komputasi (detik), ukuran model (KB), FPR, dan FNR, yang bernilai lebih rendah untuk kinerja yang lebih baik. Matriks keputusan yang terbentuk selanjutnya langsung masuk tahap normalisasi, pembobotan, dan pemeringkatan.

2.2.3. Normalisasi dan pemeringkatan

Metode MOORA digunakan untuk eksperimen ini karena kemampuan dalam menggabungkan kriteria keuntungan dan kriteria biaya melalui normalisasi rasio dan agregasi berbobot [19]. Pada matriks keputusan yang terbentuk dari hasil evaluasi model seperti (19) dan (20). Tahap setelah terbentuknya matriks keputusan adalah tahap normalisasi vektor:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}}. \tag{21}$$

Jika w_j adalah bobot kriteria ke-j, maka nilai ternormalisasi berbobot dari r_{ij} pada (21) didefinisikan sebagai

$$v_{ij} = w_j r_{ij}, \qquad \sum_{j=1}^{q} w_j = 1, \quad w_j \geq 0. \tag{22}$$

Himpunan kriteria keuntungan adalah

$$\mathcal{B} = \{\textit{Accuracy}, \textit{Precision}, \textit{Recall}, \textit{F1-score}, \texttt{ROC-AUC}\}, \tag{23}$$

sedangkan himpunan kriteria biaya adalah

$$\mathcal{C} = \{\textit{Log Loss}, \textit{Computational Time}, \textit{Model Size}, \texttt{FPR}, \texttt{FNR}\}. \tag{24}$$

Skor optimasi MOORA untuk alternatif M_i dihitung dari v_{ij} pada (22) dengan himpunan $\mathcal{B}$ pada (23) dan $\mathcal{C}$ pada (24) sebagai

$$Y_i = \sum_{j\in\mathcal{B}} v_{ij} - \sum_{j\in\mathcal{C}} v_{ij}. \tag{25}$$

Alternatif terbaik adalah model dengan nilai Y_i terbesar:

$$M^\star = \arg\max_{M_i\in\mathcal{M}} Y_i. \tag{26}$$

Bobot kriteria ditentukan secara manual berdasarkan tujuan sistem peringatan *dropout* pada mahasiswa, ketidakseimbangan kelas, risiko kesalahan klasifikasi, kualitas probabilitas prediksi, dan efisiensi implementasi. Bobot yang digunakan adalah

$$\begin{aligned} w = (&0.10,\ 0.10,\ 0.14,\ 0.14,\ 0.14,\\ &0.09,\ 0.05,\ 0.04,\ 0.08,\ 0.12), \end{aligned} \tag{27}$$

dengan urutan kriteria *Accuracy*, *Precision*, *Recall*, *F1-score*, ROC-AUC, *Log Loss*, *Computational Time*, *Model Size*, FPR, dan FNR.

Bobot *accuracy* dan *precision* keduanya ditetapkan sebesar 0,10 karena penting untuk menilai ketepatan klasifikasi umum dan ketepatan prediksi *dropout* yang benar. Tetapi, bobot keduanya tidak terlalu dominan karena akurasi dapat bias pada data tidak seimbang [12]. Bobot *recall*, F1-*score*, dan ROC-AUC masing-masing ditetapkan sebesar 0,14 karena ketiga metrik menunjukkan kemampuan model dalam mendeteksi mahasiswa berisiko *dropout* dan memisahkan antara kelas positif dari kelas negatif serta semakin banyak mahasiswa *dropout* aktual yang terdeteksi. F1-*score* digunakan untuk menyeimbangkan *precision* dan *recall*, sedangkan ROC-AUC

Tabel 2: Alternatif model yang dievaluasi beserta konfigurasi utama dan rujukan metodologis.

Kode	Model	Konfigurasi utama	Ref.
M_1	*Logistic Regression*	`penalty=l2`, `C=100`, `solver=lbfgs`, `max_iter=2000`, `class_weight=balanced`	[14]
M_2	*Gaussian Naive Bayes*	`var_smoothing=1e-9`	[15]
M_3	*K-Nearest Neighbors*	`n_neighbors=5`, `weights=distance`, `metric=minkowski`, `p=2`, `n_jobs=-1`	[16]
M_4	*Decision Tree*	`criterion=gini`, `max_depth=10`	[14]
M_5	*Random Forest*	`n_estimators=700`, `max_features=log2`, `max_depth=5`, `n_jobs=-1`	[15]
M_6	*Support Vector Machine*	`C=100`, `kernel=rbf`, `gamma=0.1`, `probability=True`	[15]
M_7	*Linear Discriminant Analysis*	`solver=lsqr`, `shrinkage=auto`	[17], [18]
M_8	*Quadratic Discriminant Analysis*	`reg_param=0.1`	[18]

Tabel 3: Kriteria dan bobot MOORA.

Kriteria	Tipe	Bobot
Accuracy	*Benefit*	0.10
Precision	*Benefit*	0.10
Recall	*Benefit*	0.14
F1-score	*Benefit*	0.14
`ROC-AUC`	*Benefit*	0.14
Log Loss	*Cost*	0.09
Computational Time (s)	*Cost*	0.05
Model Size (KB)	*Cost*	0.04
`FPR`	*Cost*	0.08
`FNR`	*Cost*	0.12

digunakan untuk menilai kemampuan diskriminatif model pada variasi ambang keputusan. Pada data tidak seimbang, `ROC-AUC` tetap perlu dikombinasikan dengan metrik berbasis *precision-recall* agar interpretasi performa tidak terlalu optimistis [20].

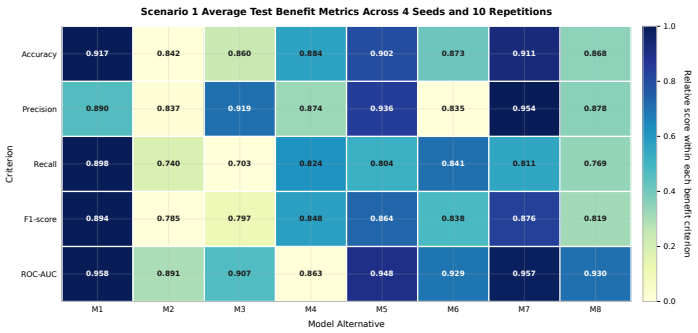
Pada kriteria biaya, *log loss* diberi bobot 0,09 karena menghitung kualitas probabilitas prediksi, baik itu label diskrit maupun kontinu. Model yang memiliki prediksi salah dengan probabilitas tinggi harus dipenalti yang lebih besar, terutama dalam sistem risiko yang menggunakan probabilitas [21]. Waktu komputasi diberi bobot 0,05 dan ukuran model diberi bobot 0,04 karena keduanya merupakan efisiensi implementasi. Beberapa model yang terlalu lambat dan berat dapat meningkatkan biaya komputasi, memori, dan penyimpanan, sehingga kurang ideal untuk sistem akademik yang perlu diperbarui secara berkala [22]. `FPR` diberi bobot 0,08 karena *false positive* dapat menyebabkan mahasiswa yang sebenarnya tidak berisiko diklasifikasikan sebaliknya, sedangkan `FNR` diberi bobot 0,12 karena *false negative* lebih kritis karena mahasiswa yang benar-benar berisiko dapat tidak terdeteksi dan kehilangan peluang memperoleh tindak lanjut akademik tepat waktu [13].

3. Hasil dan Pembahasan

Hasil eksperimen adalah *output* dari *Kaggle Notebook* yang sudah dirancang sebelumnya (Desain eksperimen)[1]. Di mana *notebook* versi 12 berjudul *Final for Publication* dengan *runtime* eksekusi penuh tercatat 1584.8s detik atau 26 menit 25 detik, dengan konfigurasi $\mathcal{M} = \{M_1, \ldots, M_8\}$ seperti model *Logistic Regression*, *Gaussian Naive Bayes*, *K-Nearest Neighbors*, *Decision Tree*, *Random Forest*, *Support Vector Machine*, *Linear Discriminant Analysis*, dan *Quadratic Discriminant Analysis* dirancang menggunakan *CPU Multi-Thread*. Model-model tersebut harus dilakukan pelatihan dan uji tes terlebih dahulu sehingga mendapatkan metrik evaluasi. Karena tidak cukup hanya satu kali percobaan dan belum bisa mengecek stabilitas model maka model dilakukan perulangan sebanyak 10 kali dengan beberapa *random seed* (0, 7, 42, 43), lalu seluruh hasil dirata-ratakan dan menjadi matriks keputusan untuk pemeringkatan model.

3.1. Performa prediktif dan biaya model

Performa prediktif dan biaya model ditampilkan pada Gambar 1 dan Gambar 2 menunjukkan rata-rata hasil uji antar-*seed* dan perulangan. *Logistic Regression* (`M1`) memperoleh akurasi tertinggi, yaitu 0.916667, dengan F1-*score* 0.893808 dan `ROC-AUC` 0.958256. *Linear Discriminant Analysis* (`M7`) memiliki presisi tertinggi sebesar 0.954426 dengan ukuran model yang sangat efisien, yaitu 16 KB dan waktu komputasi hanya 0.042808 detik. *Random Forest* (`M5`) memiliki presisi kedua tertinggi (0.935784), namun ukuran modelnya sangat besar, yaitu 3.211 KB dengan waktu komputasi 2.551991 detik. *Support Vector Machine* (`M6`) juga memerlukan waktu komputasi tinggi sebesar 2.386937 detik.



Gambar 1: Perbandingan rata-rata metrik keuntungan ($\mathcal{B}$) lintas *seed* dan repetisi untuk kedelapan alternatif model.

Dari sisi kesalahan klasifikasi yang ditunjukkan pada Tabel 5, Tabel 6, dan Gambar 2, rata-rata *confusion matrix* menunjukkan bahwa `M1` menghasilkan $TN = 410{,}50$, $FP = 31{,}50$, $FN = 29{,}00$, dan $TP = 255{,}00$. Nilai `FNR` `M1` sebesar 0.102113 adalah yang terendah di antara semua model, sehingga *Logistic Regression* paling sedikit melewatkan mahasiswa *dropout* asli. `M7` (*Linear Discriminant Analysis*) memiliki `FPR` terendah sebesar 0.024887, namun `FNR`-nya 0.189261. `M5` (*Random Forest*) memiliki `FPR` kedua terendah sebesar 0.035633, tetapi `FNR`-nya 0.196303. Dalam sistem pendukung

[1] https://www.kaggle.com/code/izzarsulynashrudin/moora-dropout

Tabel 4: Rata-rata metrik evaluasi utama pada skenario *Graduate* versus *Dropout*, didapatkan pada *output Kaggle* dengan nama *file* `scenario1_moora_aggregated_evaluation_table_overall.csv`.

Model	*Acc.*	*Prec.*	*Recall*	F1	ROC-AUC	*Log Loss*	*Time* (s)	*Size* (KB)
M1	0.916667	0.890090	0.897887	0.893808	0.958256	0.237023	0.100154	5
M2	0.841598	0.836898	0.740317	0.785111	0.891393	1.309660	0.027494	6
M3	0.859848	0.919435	0.703345	0.796622	0.907472	0.715750	0.046161	844
M4	0.884298	0.873873	0.823944	0.847743	0.863036	1.287948	0.048507	26
M5	0.901515	0.935784	0.803697	0.864497	0.948382	0.288025	2.551991	3211
M6	0.872590	0.835063	0.840669	0.837615	0.929191	0.334339	2.386937	498
M7	0.910813	0.954426	0.810739	0.876131	0.956910	0.255666	0.042808	16
M8	0.867769	0.877510	0.769366	0.819430	0.930243	0.733060	0.043807	26

keputusan akademik, FPR rendah mengurangi mahasiswa yang salah diprediksi berisiko *dropout*, tetapi FNR tinggi berarti lebih banyak mahasiswa berisiko yang terlewatkan. Oleh karena itu, M1 lebih sesuai untuk prioritas utama sistem dalam meminimalkan mahasiswa berisiko yang terlewatkan serta perbedaan tersebut juga merupakan pilihan model optimal yang tentu bergantung pada prioritas kebijakan masing-masing.

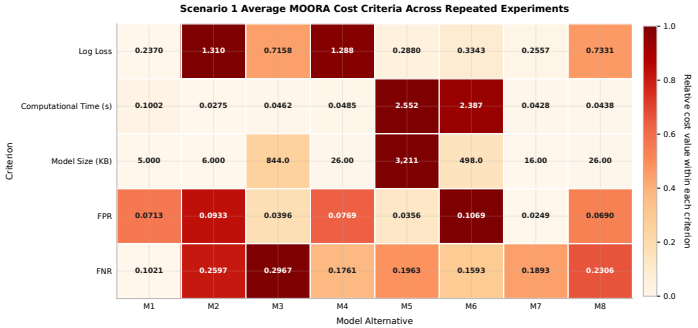
Tabel 5: *Confusion matrix components for each alternative*, didapatkan pada *output Kaggle* dengan nama *file* `scenario1_summary_mean_only_overall_model.csv`.

Alternative	***Model Name***	TN	FP	FN	TP
M1	*Logistic Regression*	410.50	31.50	29.00	255.00
M2	*Gaussian Naive Bayes*	400.75	41.25	73.75	210.25
M3	*K-Nearest Neighbors*	424.50	17.50	84.25	199.75
M4	*Decision Tree*	408.00	34.00	50.00	234.00
M5	*Random Forest*	426.25	15.75	55.75	228.25
M6	*Support Vector Machine*	394.75	47.25	45.25	238.75
M7	*Linear Discriminant Analysis*	431.00	11.00	53.75	230.25
M8	*Quadratic Discriminant Analysis*	411.50	30.50	65.50	218.50

Tabel 6: Rata-rata FPR dan FNR pada data uji, didapatkan pada *output Kaggle* dengan nama *file* `scenario1_moora_aggregated_evaluation_table_overall.csv`.

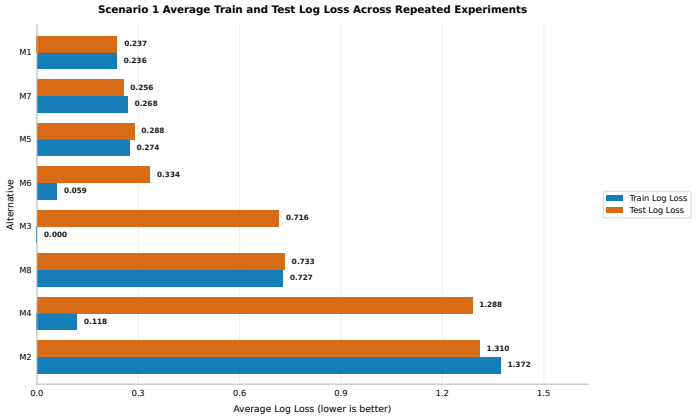
Model	FPR	FNR
M1	0.071267	0.102113
M2	0.093326	0.259683
M3	0.039593	0.296655
M4	0.076923	0.176056
M5	0.035633	0.196303
M6	0.106900	0.159331
M7	0.024887	0.189261
M8	0.069005	0.230634

Dari perspektif kualitas probabilitas, Gambar 3 memperlihatkan perbandingan rata-rata *log loss* antara data latih dan data uji. Kesenjangan kecil antara keduanya pada M1, M5, M6, M7, dan M8 menandakan bahwa model-model tersebut tidak mengalami *overfitting* yang signifikan. M1 (*Logistic Regression*) mencapai *log loss* uji terendah (0.237023), mencerminkan kalibrasi probabilitas terbaik di antara semua alternatif. Sebaliknya, M2 (*Gaussian Naive Bayes*) memiliki *log loss* uji tertinggi (1.309660), yang menandakan estimasi probabilitas tidak andal akibat asumsi independensi fitur yang terlalu kuat pada data dengan korelasi antarfitur yang nyata.



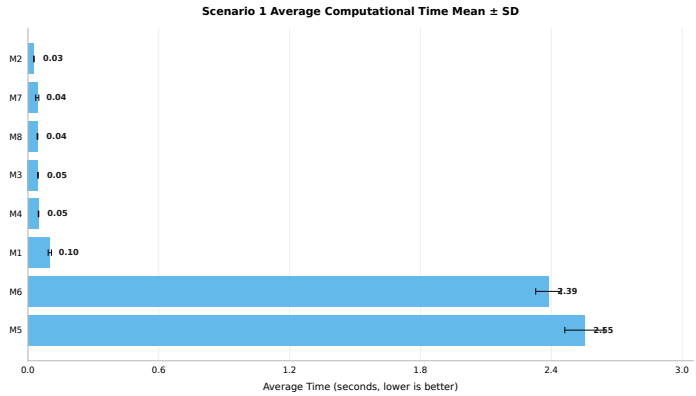
Gambar 2: Perbandingan rata-rata kriteria biaya ($\mathcal{C}$) lintas *seed* dan repetisi untuk kedelapan alternatif model.

M4 (*Decision Tree*) juga memiliki *log loss* tinggi (1.287948) karena probabilitas prediksinya bersifat deterministik pada setiap daun pohon, bukan dikalibrasi secara kontinu.

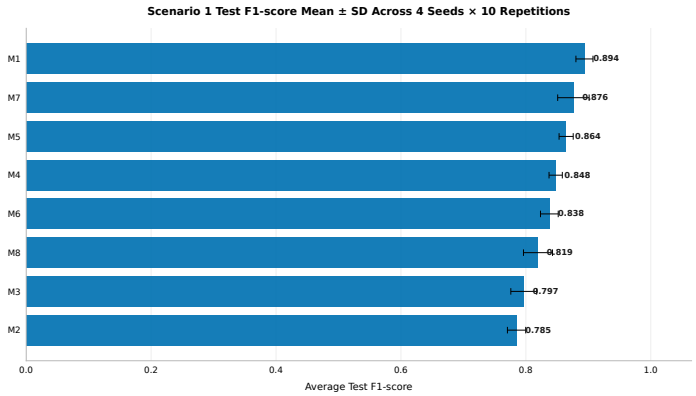


Gambar 3: Perbandingan rata-rata *log loss* pada data latih dan data uji untuk kedelapan alternatif model lintas *seed* dan repetisi.

Dari sisi efisiensi komputasi, Gambar 4 menampilkan waktu komputasi rata-rata beserta simpangan baku untuk setiap model. Model *linear* seperti M7 (*Linear Discriminant Analysis*) dan M1 (*Logistic Regression*) menunjukkan waktu komputasi rendah, masing-masing 0.042808 dan 0.100154 detik, karena kompleksitasnya berskala *linear* terhadap jumlah sampel dan fitur. Sebaliknya, M5 (*Random Forest*) dan M6 (*Support Vector Machine*) memerlukan waktu masing-masing 2.551991 dan 2.386937 detik, akibat kompleksitas agregasi 700 pohon keputusan dan optimasi *kernel radial basis function* (RBF). Perbedaan waktu komputasi ini menjadi faktor penting dalam komponen biaya MOORA.



Gambar 4: Rata-rata waktu komputasi (detik) beserta simpangan baku untuk kedelapan alternatif model pada seluruh eksperimen berulang.

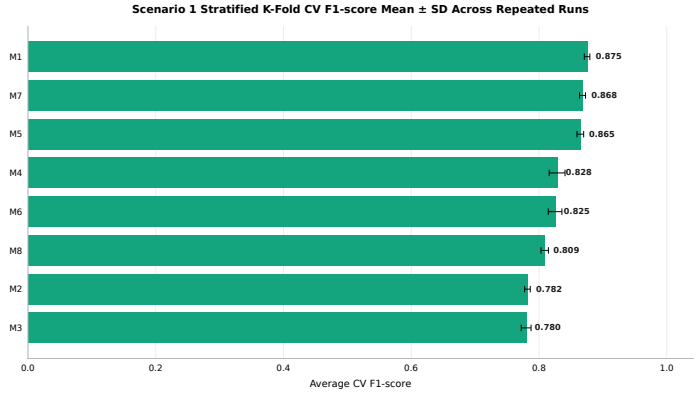


Gambar 6: Rata-rata F1-*score* pada data uji *hold-out* lintas *seed* dan repetisi, dengan *error bar* menunjukkan simpangan baku antar-*seed*.

3.2. Konsistensi model pada *Stratified K-Fold Cross-Validation*

Tahap *Stratified k-fold cross-validation* dengan $k = 5$ diimplementasikan pada subset data latih di setiap eksperimen untuk mengetahui konsistensi model dan mendeteksi potensi *overfitting* sebelum tahap evaluasi pada data uji *hold-out*. Cara kerja *k-fold cross-validation* dengan membagi data latih $\mathcal{D}_{\text{train}}^{(s)}$ menjadi $k = 5$ bagian dengan ukuran relatif sama serta tetap menjaga proporsi kelas *Dropout* dan *Graduate* di setiap bagian tetap (*stratified*). Saat iterasi ke-ℓ, satu *fold* $\mathcal{F}_\ell$ berfungsi sebagai data validasi internal sementara itu gabungan *fold* lainnya dipakai untuk melatih model. Proses ini diulang hingga k kali, lalu setiap sampel data latih berurutan untuk divalidasi tepat satu kali. Rata-rata F1-*score* dari kelima *fold* lalu menjadi indikator kemampuan model dalam mengenali pola baru.

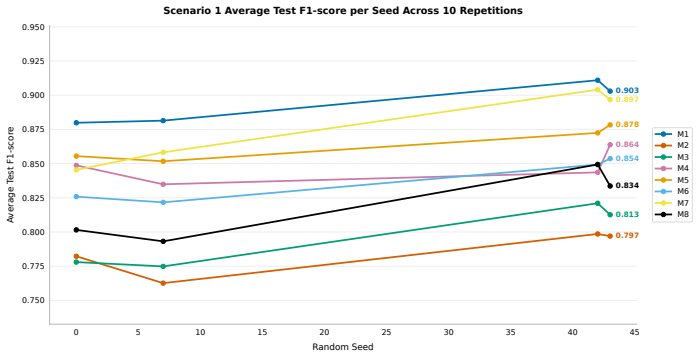
Nilai *input* pada matrik keputusan tetap memakai metrik dari data uji *hold-out*, bukan hasil dari *cross-validation*. Nilai `CV` bertujuan untuk menjadi bukti tambahan sebagai performa model pada data uji tidak bergantung pada satu konfigurasi partisi data agar model menunjukan generalisasi yang sesungguhnya. Gambar 5 menyajikan rata-rata F1-*score* `CV` dengan *error bar*, sedangkan Gambar 6 menggambarkan F1-*score* pada data uji *hold-out* yang seharusnya konsisten dengan hasil `CV` jika tidak terjadi *overfitting*.



Gambar 5: Rata-rata F1-*score* hasil *stratified k-fold cross-validation* (5 *fold*) pada data latih lintas *seed* dan repetisi, dengan *error bar* menunjukkan simpangan baku antar-*seed*.

Dari Gambar 5 dan Gambar 6 terlihat nilai `CV` F1-*score* dan *hold-out* F1-*score* pada `M1` (*Logistic Regression*) dan `M7` (*Linear Discriminant Analysis*) sangat berdekatan dengan simpangan baku kecil, menegaskan konsistensi performa yang tinggi tanpa *overfitting*. Pada model *Logistic Regression* dengan regularisasi `L2` dan `class_weight=balanced` mampu menemukan batas keputusan yang stabil walaupun terjadi variasi komposisi data latih, sementara `LDA` menggunakan asumsi distribusi *Gaussian* yang menghasilkan batas keputusan analitik yang tidak bergantung pada konfigurasi partisi tertentu. Sebaliknya, `M3` (*K-Nearest Neighbors*) menunjukkan simpangan baku lebih besar karena algoritma berbasis jarak ini sangat sensitif terhadap distribusi lokal sampel dalam data latih.

Untuk memverifikasi bahwa konsistensi performa tidak bergantung pada *seed* tertentu, Gambar 7 menyajikan rata-rata F1-*score* per *seed* secara terpisah.



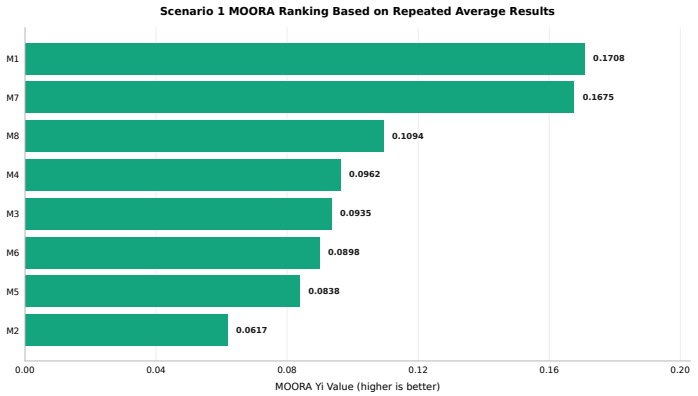
Gambar 7: Rata-rata F1-*score* per *seed* (0, 7, 42, 43) untuk kedelapan model, menggambarkan konsistensi performa lintas pembagian data yang berbeda.

Gambar 7 menunjukkan `M1` (*Logistic Regression*) secara konsisten menduduki F1-*score* tertinggi pada seluruh *seed*. Kestabilan hasil ini menunjukkan bahwa posisi unggul `M1` dalam pemeringkatan MOORA tidak hanya terjadi karena kebetulan pada satu skenario pembagian data tertentu. Sebaliknya, `M4` (*Decision Tree*) menunjukkan hasil yang lebih berubah-ubah antar-*seed*. Hal ini mengindikasikan performanya cukup sensitif terhadap perbedaan struktur pohon yang terbentuk pada tiap pembagian data. Kesimpulannya, evaluasi berbasis 4 *seed* dan 10 perulangan per *seed* (total 40 *run*) memberikan estimasi performa model yang jauh lebih bagus dibandingkan evaluasi satu *seed* saja.

3.3. Hasil pemeringkatan MOORA

Nilai pada Tabel 4 dan Tabel 6 digunakan sebagai matriks keputusan MOORA berukuran 8×10, di mana setiap elemen x_{ij} merupakan rata-rata dari 40 *run* ulangan sesuai (8). Proses MOORA meliputi normalisasi vektor, pembobotan dengan $\mathbf{w}$ yang sudah ditentukan pada (27) dan Tabel 3, serta agregasi skor optimasi sesuai (25). Hasil normalisasi vektor dirangkum pada Tabel 7.

Berdasarkan nilai Y_i, alternatif diurutkan dari nilai terbesar ke nilai terkecil sesuai (26). Dengan demikian, *Logistic Regression* menjadi model terbaik berdasarkan perhitungan MOORA berbobot. Hasil lengkap dari perhitungan ini kemudian diringkas pada Tabel 8.



Gambar 8: Hasil pemeringkatan MOORA berbobot berdasarkan rata-rata 40 eksperimen (4 *seed* x 10 perulangan per *seed*).

Hasil pada Tabel 8 dan Gambar 8 menolak anggapan bahwa model paling kompleks otomatis lebih optimal. *Random Forest* (`M5`) memiliki presisi tertinggi (0.935784), tetapi penalti terhadap ukuran model (3.211 KB) dan waktu komputasi (2.551991 detik) menurunkan posisinya ke peringkat ketujuh. *Support Vector Machine* (`M6`) juga terjadi penurunan ke peringkat keenam karena waktu komputasi 2.386937 detik. Sebaliknya, `M7` (*Linear Discriminant Analysis*) menduduki peringkat kedua dengan waktu komputasi yang lebih singkat hanya 0.042808 detik dan ukuran model 16 KB meskipun `FNR`-nya 0.189261. Secara matematis, penalti ini masuk melalui kriteria $\sum_{j \in \mathcal{C}} v_{ij}$, sehingga model prediktif tinggi tetap dapat kalah jika biaya *deployment*-nya besar. Ini penting untuk sistem deteksi yang membutuhkan model ringan, dapat diulang secara periodik, dan mudah diintegrasikan ke sistem informasi akademik.

Gaussian Naive Bayes (`M2`) berada pada peringkat terakhir karena kombinasi performa prediktif rendah dan *log loss* tertinggi sebesar 1.309660, menunjukkan probabilitas prediksi yang tidak stabil. *K-Nearest Neighbors* (`M3`) juga tidak kompetitif karena *recall* sebesar 0.703345 dan `FNR` sebesar 0.296655, sehingga gagal mendeteksi proporsi *dropout* sebenarnya yang relatif besar.

3.4. Skenario prediksi mahasiswa *Enrolled*

Pada skenario ini, seluruh delapan alternatif model dilatih ulang menggunakan data *Graduate* versus *Dropout* dengan satu *seed* tunggal, yaitu *seed* 42, dan diulang sebanyak sepuluh kali. Berbeda dengan Skenario 1 yang menggunakan 4 *seed* dengan 10 perulangan per *seed* (total 40 *run* per model), Skenario 2 menggunakan satu *seed* saja sehingga menghasilkan sepuluh *run* per model. Pilihan *seed* 42 didasarkan pada kestabilan distribusi data uji hasil stratifikasi. Karena model deterministik seperti *Logistic Regression*, *Linear Discriminant Analysis*, dan *Quadratic Discriminant Analysis* menghasilkan *output* inferensi yang identik pada setiap repetisi dengan *seed* yang sama, nilai rata-rata kesepuluh perulangan pada tabel ini identik dengan nilai per repetisi individual.

Setelah pelatihan, setiap model diterapkan langsung pada seluruh 794 data mahasiswa *Enrolled* yang belum memiliki label akhir. Hasil Tabel 9 merangkum prediksi risiko *dropout* dari kedelapan model, sedangkan Tabel 10 memfokuskan pada tiga model teratas MOORA. Karena label akhir *Enrolled* belum tersedia, angka ini tidak boleh ditafsirkan sebagai akurasi, melainkan sebagai estimasi prioritas intervensi.

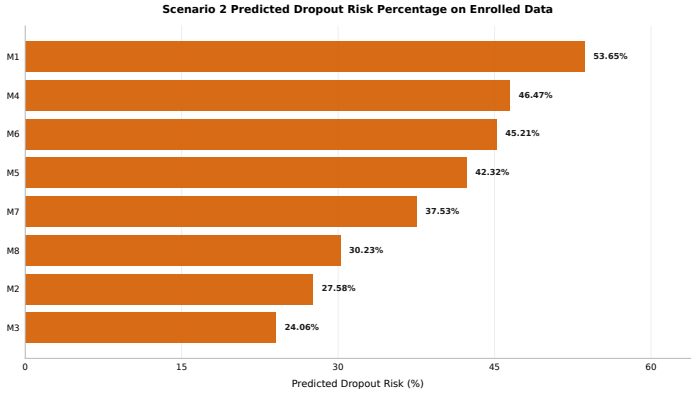
Hasil Tabel 10 memfokuskan pada tiga model teratas MOORA. *Logistic Regression* memprediksi 426 dari 794 mahasiswa sebagai risiko *dropout* (53.65%), *Linear Discriminant Analysis* memprediksi 298 mahasiswa (37.53%), dan *Quadratic Discriminant Analysis* memprediksi 240 mahasiswa (30.23%).

Perbedaan estimasi risiko antara `M1`, `M4`, dan `M7` menunjukkan keputusan tidak seharusnya hanya pada label biner tunggal. Secara operasional, probabilitas $\hat{p}_i$ lebih informatif karena memiliki potensi pemeringkatan mahasiswa berdasarkan tingkat risiko. Misalnya, institusi memilih kuantil atas probabilitas risiko sebagai konseling akademik. Jika kapasitas keputusan K mahasiswa, maka himpunan prioritas dapat ditulis sebagai

$$\mathcal{I}_K = \mathrm{TopK}_i\left(\hat{p}_i\right), \tag{28}$$

bukan seluruh mahasiswa dengan $\hat{p}_i \geq 0.5$. Hal ini, sesuai (28), lebih adaptif terhadap kapasitas layanan akademik dan menghindari klasifikasi yang terlalu stagnan.

Gambar 9 menampilkan persentase mahasiswa *Enrolled* yang diprediksi berisiko *dropout* oleh setiap model, sedangkan Gambar 10 menampilkan rata-rata probabilitas risiko yang dihasilkan antarmodel. Perbedaan antarmodel tersebut memperlihatkan perbedaan mendasar, yaitu setiap algoritma memiliki batas keputusan dan prediksi probabilitas kelas.



Gambar 9: Persentase mahasiswa *Enrolled* yang diprediksi sebagai risiko *dropout* oleh kedelapan model (Skenario 2, *seed* 42, rata-rata 10 repetisi).

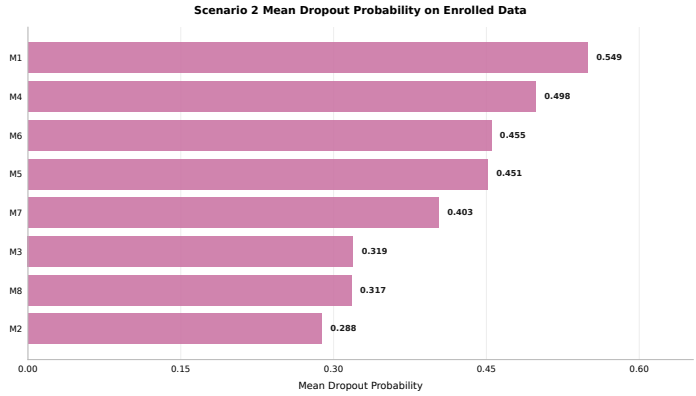
Gambar 11 menunjukkan persentase prediksi risiko *dropout* antara tiga model teratas MOORA (`M1`, `M7`, `M8`) dan model-model berperingkat rendah. Tiga model terbaik menghasilkan rentang prediksi kelas positif yang bervariasi antara 30,23%–53,65%, sedangkan beberapa model berperingkat rendah

Tabel 7: Hasil ringkasan normalisasi vektor rata-rata MOORA pada skenario *Graduate* versus *Dropout*, sesuai dengan *output Kaggle* dengan nama *file* `scenario1_moora_normalized_matrix_overall_average.csv`.

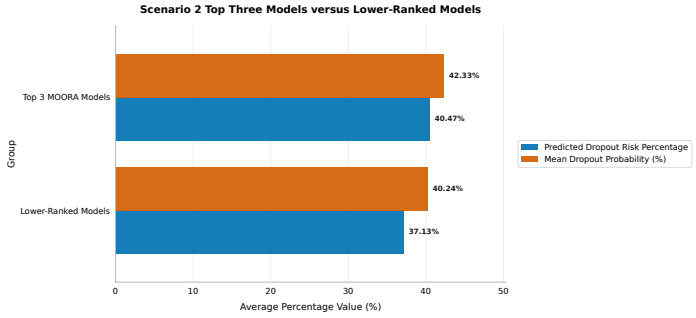
Model	*Acc.*	*Prec.*	*Recall*	F1	ROC-AUC	*Log Loss*	*Time* (s)	*Size* (KB)	FPR	FNR
M1	0.367354	0.353064	0.396438	0.375809	0.366800	0.108869	0.028640	0.001489	0.359257	0.172653
M2	0.337270	0.331965	0.326867	0.330106	0.341206	0.601548	0.007862	0.001787	0.470456	0.439073
M3	0.344584	0.364704	0.310543	0.334946	0.347361	0.328756	0.013200	0.251381	0.199587	0.501585
M4	0.354382	0.346632	0.363790	0.356440	0.330352	0.591575	0.013871	0.007744	0.387770	0.297677
M5	0.361282	0.371189	0.354851	0.363485	0.363021	0.132295	0.729762	0.956378	0.179629	0.331910
M6	0.349690	0.331238	0.371174	0.352182	0.355675	0.153567	0.682563	0.148326	0.538886	0.269397
M7	0.365008	0.378584	0.357960	0.368376	0.366285	0.117432	0.012241	0.004766	0.125455	0.320002
M8	0.347758	0.348074	0.339693	0.344536	0.356077	0.336706	0.012527	0.007744	0.347852	0.389957

Tabel 8: Peringkat akhir MOORA berdasarkan rata-rata eksperimen berulang.

Rank	Model	*Benefit Sum*	*Cost Sum*	Y_i	*Top 3*
1	M1 *Logistic Regression*	0.231508	0.060749	0.170760	Ya
2	M7 *Linear Discriminant Analysis*	0.227326	0.059808	0.167518	Ya
3	M8 *Quadratic Discriminant Analysis*	0.215226	0.105863	0.109363	Ya
4	M4 *Decision Tree*	0.217183	0.120988	0.096195	Tidak
5	M3 *K-Nearest Neighbors*	0.209928	0.116460	0.093467	Tidak
6	M6 *Support Vector Machine*	0.219157	0.129321	0.089836	Tidak
7	M5 *Random Forest*	0.224637	0.140849	0.083788	Tidak
8	M2 *Gaussian Naive Bayes*	0.206669	0.144929	0.061739	Tidak



Gambar 10: Rata-rata probabilitas risiko *dropout* untuk mahasiswa *Enrolled* yang dihasilkan oleh kedelapan model pada Skenario 2.



Gambar 11: Perbandingan persentase prediksi risiko *dropout* antara tiga model teratas MOORA (M1, M7, M8) dan model berperingkat lebih rendah pada 794 data *Enrolled*.

seperti M4 (*Decision Tree*, 46,47%) dan M6 (*Support Vector Machine*, 45,21%) menghasilkan prediksi lebih tinggi. Namun, dengan *log loss* yang besar pada ketiga model tersebut menjadikan hasil kurang meyakinkan. Hal ini memperkuat bahwa model terbaik MOORA bukan semata-mata yang menghasilkan prediksi kelas positif tertinggi, melainkan model yang dapat berkalibrasi serta dipertanggungjawabkan secara probabilistik.

3.5. Evaluasi

Secara keseluruhan, hasil eksperimen pemeringkatan model berbasis MOORA berbobot menghasilkan urutan yang berbeda dibandingkan dengan pemeringkatan berbasis metrik tunggal. Keunggulan M1 (*Logistic Regression*) bukan karena akurasi tertinggi (0.916667), melainkan keseimbangan konsisten antara 10 kriteria. Konsistensi performa M1 pada keempat *seed* dengan simpangan baku F1-*score* yang relatif kecil disajikan pada Gambar 5 dan Gambar 6 mengonfirmasi hasil eksperimen ini bukan kebetulan dari satu partisi data, tetapi kemampuan model dalam mengenali pola secara umum. M7 (*Linear Discriminant Analysis*) di peringkat kedua membuktikan model statistik berbasis distribusi *Gaussian* dapat bersaing dengan model yang lebih kompleks, terutama pada efisiensi komputasi (0.042808 detik, 16 KB) dan presisi tertinggi (0.954426). M8 (*Quadratic Discriminant Analysis*) di peringkat ketiga menunjukkan fleksibilitas batas keputusan *nonlinear* dari QDA memberikan nilai plus dengan biaya komputasi rendah.

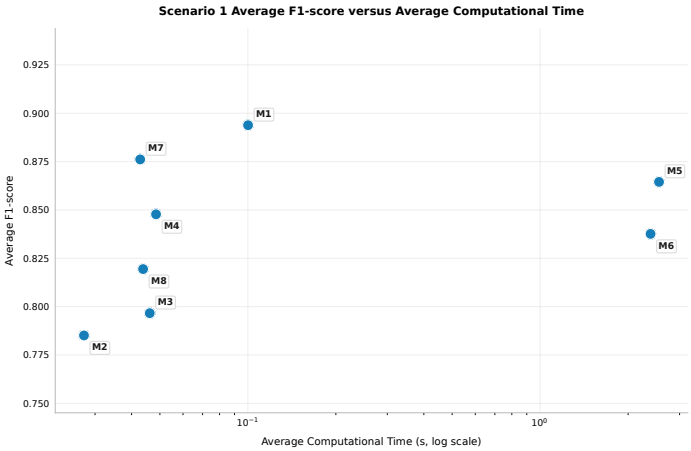
Penurunan peringkat M5 (*Random Forest*) dari posisi unggul pada presisi (peringkat kedua, 0.935784) menjadi peringkat ketujuh disebabkan penalti biaya. Ukuran model sebesar 3.211 KB dan waktu komputasi 2.551991 detik menciptakan skor biaya yang tidak bisa ditanggung oleh keunggulan prediktifnya. Hal tersebut juga terjadi pada M6 (*Support Vector Machine*) yang mengalami penurunan peringkat akibat waktu komputasi 2.386937 detik yang bisa dilihat pada Gambar 12. Pola semacam ini menunjukkan sistem peringatan dini akademik perlu diperbarui secara berkala, termasuk efisiensi *deployment* yang tidak dapat diabaikan [22]. M2 (*Gaussian Naive Bayes*) berada di peringkat terakhir karena memiliki kriteria biaya *log loss* tertinggi (1.309660) dan performa metrik evaluasi

Tabel 9: Prediksi risiko *dropout* pada 794 data *Enrolled* untuk seluruh delapan model (*seed* 42, rata-rata 10 repetisi).

Rank	Kode	Model	Risiko *Dropout*	*Graduate-like*	Risiko (%)	Rerata Prob.	Median Prob.
1	`M1`	*Logistic Regression*	426	368	53.65	0.5492	0.5587
2	`M7`	*Linear Discriminant Analysis*	298	496	37.53	0.4032	0.3039
3	`M8`	*Quadratic Discriminant Analysis*	240	554	30.23	0.3175	0.0317
4	`M4`	*Decision Tree*	369	425	46.47	0.4984	0.1875
5	`M3`	*K-Nearest Neighbors*	191	603	24.06	0.3191	0.2120
6	`M6`	*Support Vector Machine*	359	435	45.21	0.4547	0.4376
7	`M5`	*Random Forest*	336	458	42.32	0.4513	0.3985
8	`M2`	*Gaussian Naive Bayes*	219	575	27.58	0.2884	0.0113

Tabel 10: Prediksi risiko *dropout* pada data *Enrolled* untuk tiga model teratas MOORA.

Rank MOORA	Model	Risiko *Dropout*	*Graduate-like*	Risiko (%)	Rerata Probabilitas
1	`M1` *Logistic Regression*	426	368	53.6524	0.5492
2	`M7` *Linear Discriminant Analysis*	298	496	37.5315	0.4032
3	`M8` *Quadratic Discriminant Analysis*	240	554	30.2267	0.3175



Gambar 12: *Trade-off* antara F1-*score* dan waktu komputasi untuk kedelapan alternatif model. Model di kuadran kanan atas memiliki F1-*score* tinggi, tetapi waktu komputasi besar.

rendah secara keseluruhan, disebabkan oleh setiap fitur saling independen, padahal struktur data penelitian ini memiliki korelasi antarfitur.

Terdapat empat hal yang perlu diperhatikan pada penelitian ini. Pertama, MOORA berbobot memberikan pemeringkatan yang lebih sesuai untuk seleksi model nyata dibandingkan dengan pemilihan berdasarkan satu metrik.

Kedua, prediksi pada mahasiswa *Enrolled* adalah simulasi risiko, bukan validasi eksternal. Tanpa adanya label *dropout* atau *graduate*, tidak ada estimasi *unbiased* terhadap akurasi.

Ketiga, penelitian ini memiliki beberapa keterbatasan. (1) Pembobotan MOORA ditentukan secara manual berdasarkan pertimbangan domain, tanpa diturunkan secara *data-driven* yang berpotensi memengaruhi urutan pemeringkatan. (2) Alternatif model terbatas pada delapan model klasifikasi *machine learning* klasik dan tidak mencakup pendekatan *deep learning* yang dapat menangkap pola temporal pada riwayat akademik mahasiswa. (3) Seluruh eksperimen dilakukan pada *dataset* yang diambil dari satu institusi, sehingga jika diterapkan pada *dataset* yang memiliki karakteristik berbeda memerlukan validasi ulang. (4) Ukuran model yang diukur hanyalah *file size*, belum mewakili konsumsi memori saat model berjalan langsung.

Keempat, arah untuk penelitian selanjutnya. Pembobotan berbasis preferensi *stakeholder* seperti metode *Analytic Hierarchy Process* (`AHP`) atau *Best-Worst Method* (`BWM`) yang dapat mengurangi kecondongan bobot tertentu dan menghasilkan pembobotan yang dapat dipertanggungjawabkan secara prosedural. Ekspansi alternatif model ke arah *gradient boosting* (*XGBoost*, *LightGBM*) dan *deep learning* (`LSTM`, *Transformer*) yang memperluas lingkup dan menguji kompleksitas model yang lebih tinggi secara multikriteria. Terakhir, dapat dilakukan pengembangan sistem *early warning* berbasis model terbaik yang dipilih dan dapat divalidasi ulang pada mahasiswa tahun akademik berikutnya untuk memberikan bukti efektivitas model dalam pengambilan keputusan di kondisi praktik nyata.

4. Kesimpulan

Penelitian ini membuktikan bahwa seleksi model prediksi *dropout* mahasiswa lebih tepat diposisikan sebagai masalah pengambilan keputusan multikriteria daripada pemilihan berbasis metrik tunggal. Dengan menggunakan metode MOORA berbobot pada delapan alternatif model dan sepuluh kriteria yang terdiri atas lima kriteria keuntungan (akurasi, presisi, *recall*, F1-*score*, `ROC-AUC`) dan lima kriteria biaya (*log loss*, waktu komputasi, ukuran model, `FPR`, `FNR`), *Logistic Regression* (`M1`) terpilih sebagai model terbaik dengan nilai optimasi $Y_1 = 0.170760$, diikuti *Linear Discriminant Analysis* (`M7`) dengan $Y_7 = 0.167518$ di peringkat kedua, dan *Quadratic Discriminant Analysis* (`M8`) dengan $Y_8 = 0.109363$ di peringkat ketiga. Keunggulan *Logistic Regression* tidak berasal dari hasil prediksinya yang tinggi, tetapi karena keseimbangan optimal antara kualitas prediksi dan efisiensi komputasi. Model ini mencapai akurasi 0.916667, F1-*score* 0.893808, `ROC-AUC` 0.958256, dan `FNR` terendah sebesar 0.102113, dengan waktu komputasi hanya 0.100154 detik serta ukuran model relatif kecil sebesar 5 KB. Sebaliknya, model yang memiliki presisi lebih tinggi mendapat penalti besar dalam MOORA, seperti *Random Forest* (`M5`) dengan ukuran model 3.211 KB dan waktu komputasi yang lambat sehingga turun ke peringkat ketujuh, dan *Support Vector Machine* (`M6`) dengan waktu komputasi 2.386937 detik menduduki peringkat keenam. *Gaussian Naive Bayes* (`M2`) berada di peringkat terakhir akibat kom-

binasi *log loss* tertinggi sebesar 1.309660 dan performa yang kurang, di mana *recall* sebesar 0.740317 menjadi terendah kedua di antara semua alternatif. Hasil tersebut membuktikan kompleksitas model tidak selalu berbanding lurus dengan kualitas pengambilan keputusan pada sistem peringatan dini akademik.

Pada skenario inferensi mahasiswa *Enrolled*, tiga model teratas MOORA memprediksi antara 30.23% hingga 53.65% dari 794 mahasiswa aktif berisiko *dropout*, yaitu `M1` memprediksi 426 mahasiswa (53.65%), `M7` memprediksi 298 mahasiswa (37.53%), dan `M8` memprediksi 240 mahasiswa (30.23%). Perbedaan hasil prediksi antarmodel ini menjadi penting dalam penggunaan nilai $\hat{p}_i$ secara kontinu daripada label biner.

Untuk implementasi lebih lanjut, model terbaik digunakan untuk sistem pendukung keputusan yang menghasilkan skor risiko yang kontinu, bukan keputusan bersifat deterministik. Sebelum *deployment*, perlu penyesuaian dan validasi terus-menerus pada data tahun akademik berikutnya sesuai prioritas kebijakan institusi masing-masing.

Pernyataan Kontribusi Penulis (CRediT)

Izzar Suly Nashrudin: Konseptualisasi, Metodologi, Analisis Formal, Kurasi Data, Penulisan Draf Awal, Penulisan Telaah dan Penyuntingan, Validasi. **Elif Thoyyibah Rahmawati:** Penulisan Draf Awal, Penulisan Telaah dan Penyuntingan. **Nur Intan Aprilia:** Penyuntingan dan Validasi. **Siti Syarifah Jamalia:** Penyuntingan dan Validasi. **Dzikry Fikrillah:** Penyuntingan dan Validasi. **Risma Muslimah:** Penyedia Lingkungan Komputasi, Pengembangan Kode.

Deklarasi Penggunaan AI atau Teknologi Berbasis AI

Penulis menyatakan bahwa teknologi *AI generatif* digunakan secara terbatas sebagai alat bantu dalam proses penyusunan naskah dan pengembangan kode. *Claude Pro* dengan model *Opus 4.8 High* digunakan untuk membantu pembuatan dan pengembangan kode program. *ChatGPT Plus* dengan fitur *Deep Research* dan model *GPT-5.5 Thinking* digunakan untuk membantu eksplorasi literatur dan perbaikan kesalahan penulisan naskah.

Seluruh isi yang dihasilkan oleh teknologi *AI* tersebut telah ditelaah, diverifikasi, dan disesuaikan ulang oleh penulis, khususnya pada aspek ketepatan substansi, struktur akademik, konsistensi istilah, dan gaya bahasa. Keputusan akhir terkait isi, interpretasi, analisis, dan kesimpulan naskah sepenuhnya berada di bawah tanggung jawab penulis.

Deklarasi Konflik Kepentingan

Penulis menyatakan tidak ada konflik kepentingan.

Pendanaan dan Ucapan Terima Kasih

Penelitian ini tidak menerima pendanaan dan kami mengucapkan terima kasih kepada Bapak Juhari, M.Si., atas bimbingan dan dukungan selama proses penelitian ini.

Ketersediaan Data

Data yang digunakan dalam penelitian ini merupakan *dataset* publik *Predict Students' Dropout and Academic Success* yang tersedia melalui UCI *Machine Learning Repository* dengan *accession ID* 697. *Dataset* dapat diakses melalui repositori UCI pada DOI `https://doi.org/10.24432/C5MC89` atau melalui laman *dataset* UCI *Machine Learning Repository*[2]. *Dataset* ini berlisensi *Creative Commons Attribution 4.0 International* (`CC BY 4.0`), sehingga dapat digunakan, dibagikan, dan diadaptasi dengan atribusi yang sesuai. Tidak terdapat data primer tambahan yang dikumpulkan oleh penulis dalam penelitian ini.

Daftar Pustaka

[1] B. Duro, A. Gomes, F. B. Correia, A. R. Borges, and J. Bernardino, "Machine learning and deep learning for dropout prediction in higher education: A review," *Computers*, vol. 15, no. 3, p. 164, 2026. DOI: 10.3390/computers15030164

[2] J. Berens, K. Schneider, S. Görtz, S. Oster, and J. Burghoff, "Early detection of students at risk: Predicting student dropouts using administrative student data from german universities and machine learning methods," *Journal of Educational Data Mining*, vol. 11, no. 3, pp. 1–41, 2019. DOI: 10.5281/zenodo.3594771

[3] O. Ovtšarenko, "Using AI to forecast student dropout risk in technical education using a learning analytics approach," *Scientific Reports*, vol. 16, p. 14 616, 2026. DOI: 10.1038/s41598-026-44919-1

[4] R. D. Romdhoni, N. W. A. Majid, and A. F. Permatasari, "Machine learning-based early warning for student dropout: Evidence from LMS behavioral engagement patterns in online higher education," *Journal of Informatics and Vocational Education*, vol. 9, no. 2, pp. 55–66, Jul. 2026. DOI: 10.20961/joive.v9i2.3508

[5] D. Andrade-Girón et al., "Predicting student dropout based on machine learning and deep learning: A systematic review," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 10, no. 5, 2023. DOI: 10.4108/eetsis.3586

[6] E. A. Adali and A. T. Isik, "The multi-objective decision making methods based on MULTIMOORA and MOOSRA for the laptop selection problem," *Journal of Industrial Engineering International*, vol. 13, no. 2, pp. 229–237, 2017. DOI: 10.1007/s40092-016-0175-5

[7] S. Chakraborty, H. N. Datta, K. Kalita, and S. Chakraborty, "A narrative review of multi-objective optimization on the basis of ratio analysis (MOORA) method in decision making," *OPSEARCH*, vol. 60, no. 4, pp. 1844–1887, 2023. DOI: 10.1007/s12597-023-00676-7

[8] N. Hidayat, L. Afuan, and H. R. Jannah, "Prescriptive learning analytics for student dropout: Integrating temporal velocity and counterfactual explanations in longitudinal data," *Journal of Computing Theories and Applications*, vol. 3, no. 4, pp. 627–646, 2026. DOI: 10.62411/jcta.15920

[9] S. Leelaluk, C. Tang, V. Švábenský, and A. Shimada, "Knowledge distillation in RNN-attention models for early prediction of student performance," in *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing*, ser. SAC '25, New York, NY, USA: Association for Computing Machinery, 2025. DOI: 10.1145/3672608.3707805

[10] M. N. Razooqee and M. A. H. Ashour, "Application of multi-objective optimization on the basis of ratio analysis (MOORA) method," *Journal of Economics and Administrative Sciences*, vol. 26, no. 122, pp. 412–421, 2020. DOI: 10.33095/jeas.v26i122.2016

[11] V. Realinho, M. Vieira Martins, J. Machado, and L. Baptista, *Predict Students' Dropout and Academic Success*, UCI Machine Learning Repository, [Dataset], 2021. DOI: 10.24432/C5MC89

[2] `https://doi.org/10.24432/C5MC89`

[12] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009. DOI: 10.1109/TKDE.2008.239

[13] S. Lee and J. Y. Chung, "The machine learning-based dropout early warning system for improving the performance of dropout prediction," *Applied Sciences*, vol. 9, no. 15, p. 3093, 2019. DOI: 10.3390/app9153093

[14] C. H. Cho, Y. W. Yu, and H. G. Kim, "A study on dropout prediction for university students using machine learning," *Applied Sciences*, vol. 13, no. 21, p. 12 004, 2023. DOI: 10.3390/app132112004

[15] H. Chavez, B. Chavez-Arias, S. Contreras-Rosas, J. M. Alvarez-Rodríguez, and C. Raymundo, "Artificial neural network model to predict student performance using nonpersonal information," *Frontiers in Education*, vol. 8, p. 1 106 679, 2023. DOI: 10.3389/feduc.2023.1106679

[16] Y. Efendi, "Machine learning-based classification of student adaptability in online learning with feature engineering," *TIERS Information Technology Journal*, vol. 6, no. 1, pp. 129–143, 2025. DOI: 10.38043/tiers.v6i1.6806

[17] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936. DOI: 10.1111/j.1469-1809.1936.tb02137.x

[18] J. H. Friedman, "Regularized discriminant analysis," *Journal of the American Statistical Association*, vol. 84, no. 405, pp. 165–175, 1989. DOI: 10.1080/01621459.1989.10478752

[19] W. K. M. Brauers and E. K. Zavadskas, "The MOORA method and its application to privatization in a transition economy," *Control and Cybernetics*, vol. 35, no. 2, pp. 445–469, 2006.

[20] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLOS ONE*, vol. 10, no. 3, e0118432, 2015. DOI: 10.1371/journal.pone.0118432

[21] T. Gneiting and A. E. Raftery, "Strictly proper scoring rules, prediction, and estimation," *Journal of the American Statistical Association*, vol. 102, no. 477, pp. 359–378, 2007. DOI: 10.1198/016214506000001437

[22] D. Preuveneers, I. Tsingenopoulos, and W. Joosen, "Resource usage and performance trade-offs for machine learning models in smart environments," *Sensors*, vol. 20, no. 4, p. 1176, 2020. DOI: 10.3390/s20041176