Angkasa Jurnal Ilmiah Bidang Teknologi
ISSN : 2085-9503 (Print)
ISSN : 2581-1355 (On Line)
Accredited Third Grade by Kemdiktisaintek, Decree No: 10/C/C3/DT.05.00/2025
DOI : 10.28989/angkasa.v18i1.3841

An End-to-End Machine Learning Pipeline for Online Purchase Intention Prediction Using Random Forest and MLOps Practices

Akas Bagus Setiawan[1,*], Hendra Yufit Riskiawan[2], Hermawan Arief Putranto[3], Taufiq Rizaldi[4], Rachmad Andri Atmoko[5]
[1,2,3,4]Department of Information Technology, Politeknik Negeri Jember, Indonesia
[5]Faculty of Vocational Studies, Universitas Brawijaya, Indonesia

Article Info

Article history:

Received January 20, 2026
Accepted February 06,2026
Published February 20, 2026

Keywords:

Purchase Intention Prediction,
Random Forest,
Machine Learning,
MLOps,
MLflow

ABSTRACT

Predicting online shoppers' purchase intention is a key issue in e-commerce because it directly affects conversion and marketing effectiveness. The main focus of this article is a Random Forest purchase-intention model accompanied by an end-to-end MLOps implementation to ensure production readiness. The dataset used is Online Shoppers Intention with 12,330 samples and 18 features representing administrative, informational, and product-related characteristics, along with behavioral metrics. Preprocessing includes missing-value imputation, numerical feature standardization, categorical feature encoding, and outlier removal using the z-score method. The model is optimized with GridSearchCV and 3-fold cross-validation. Test results show 91.38% accuracy with 73.60% precision, 56.64% recall, and 64.02% F1-score for the positive class. MLOps implementation uses MLflow for experiment tracking, Prometheus-Grafana for monitoring, and a GitHub Actions-based CI/CD pipeline for deployment automation. Overall, the Random Forest model delivers strong predictive performance on e-commerce data and is supported by an MLOps pipeline that improves reproducibility, deployment, and production monitoring.

Corresponding Author:

Akas Bagus Setiawan,
Department of Information Technology,
Politeknik Negeri Jember,
Mastrip Road, Jember, East Java 68101, Indonesia.
Email: *akasbagus_s@polije.ac.id

1. PENGANTAR

Perkembangan e-commerce yang cepat menuntut pemahaman perilaku konsumen secara lebih mendalam agar tingkat konversi meningkat. Permasalahan utamanya adalah bagaimana memprediksi niat pembelian pengunjung secara akurat berdasarkan pola browsing, sehingga keputusan bisnis seperti personalisasi dan strategi pemasaran dapat dioptimalkan.

Beragam penelitian bertema machine learning pipeline pada domain e-commerce telah membahas arsitektur pipeline, personalisasi real-time, prediksi nilai pelanggan, dan perilaku pembelian lintas konteks, namun umumnya belum fokus pada integrasi prediksi niat pembelian yang siap dioperasionalkan secara end-to-end [1], [2], [3]; [4], [5]. Pada sisi lain, studi prediksi niat pembelian online banyak menekankan pemodelan perilaku dan faktor niat, tetapi masih terbatas pada aspek prediksi tanpa penguatan pipeline produksi yang terukur [6], [7], [8];[9], [10].

Kesenjangan lainnya terlihat pada studi-studi yang menekankan ensemble, boosting, atau faktor pemasaran dan sosial media, yang meski meningkatkan akurasi, belum menunjukkan integrasi menyeluruh ke proses operasional maupun observabilitas [11], [12], [13]; [14], [15]. Di sisi algoritmik, penelitian Random Forest lebih banyak membahas variasi metode atau penerapan di domain lain, sehingga masih terbuka ruang

untuk pemanfaatan Random Forest yang terintegrasi dalam pipeline prediksi niat pembelian e-commerce [16], [17], [18]; [19], [20]. Literatur MLOps sudah memaparkan arsitektur, transisi ke LLMOps, dan isu efisiensi energi, namun belum banyak yang mengaitkan langsung praktik MLOps dengan skenario prediksi niat pembelian online secara end-to-end [21], [22], [23]; [24], [25].

Solusi yang ditawarkan pada penelitian ini adalah membangun model prediksi niat pembelian online shopper berbasis Random Forest sekaligus menerapkan pipeline MLOps end-to-end agar proses pelatihan, pelacakan eksperimen, deployment, dan monitoring berjalan terotomasi. Dengan demikian, fokus dan kontribusi utama artikel ini bersifat ganda: model prediksi yang akurat dan bukti implementasi MLOps yang operasional. Tujuan penelitian ini adalah menghasilkan model yang akurat sekaligus siap dioperasionalkan melalui praktik MLOps yang terukur.

Kebaharuan penelitian ini terletak pada integrasi menyeluruh antara model Random Forest untuk prediksi niat pembelian online shopper dan pipeline MLOps end-to-end yang menekankan reproducibility, deployment, dan monitoring produksi pada satu alur eksperimen yang utuh.

Penelitian ini disusun sebagai berikut: Bagian 2 menjelaskan metodologi penelitian termasuk preprocessing data, pemilihan model, dan implementasi MLOps. Bagian 3 menyajikan hasil dan pembahasan eksperimen. Bagian 4 memberikan kesimpulan serta implikasi.

2. METODE PENELITIAN

Penelitian ini menerapkan pipeline machine learning end-to-end yang menggabungkan preprocessing data, pengembangan Random Forest dengan hyperparameter tuning, evaluasi komprehensif, serta implementasi MLOps untuk deployment dan monitoring produksi. Metodologi dirancang untuk mengatasi ketidakseimbangan dataset khas e-commerce sekaligus menjaga reproducibility dan skalabilitas melalui praktik DevOps-MLOps.

2.1. Dataset

Dataset yang digunakan adalah Online Shoppers Intention dari Kaggle. Dataset berbentuk tabular dengan 12.330 sesi, 18 fitur prediktor, dan 1 variabel target (Revenue). Fitur meliputi aspek administratif (Administrative, Administrative_Duration), informasional (Informational, Informational_Duration), produk (ProductRelated, ProductRelated_Duration), metrik perilaku (BounceRates, ExitRates, PageValues), fitur kontekstual (SpecialDay, Month, OperatingSystems, Browser, Region, TrafficType, VisitorType, Weekend), serta fitur turunan (Administrative_Duration_Bin). Variabel target Revenue bersifat biner yang menunjukkan pembelian (1) atau tidak (0). Distribusi kelas tidak seimbang dengan dominasi non-pembeli, sehingga evaluasi model memerlukan perhatian khusus.

2.2. Keterhubungan DevOps dan MLOps

Eksperimen ini mengintegrasikan MLOps dengan praktik DevOps pada level otomatisasi dan operasi. DevOps berfokus pada siklus hidup perangkat lunak (build-test-release-deploy-operate), sedangkan MLOps memperluasnya dengan komponen khas ML seperti data/fitur, eksperimen, artefak model, dan monitoring performa [21]. Pada proyek ini, keterhubungan tersebut diwujudkan melalui versioning (GitHub), otomasi pelatihan (GitHub Actions), tracking eksperimen dan artefak (DagsHub/MLflow), containerization (Docker), serta observability (Prometheus-Grafana) [25].

Gambar 1. Keterhubungan DevOps dan MLOps

Gambar 1 menegaskan bahwa praktik DevOps dan MLOps saling terkait. Bagian DevOps menguraikan landasan pengiriman perangkat lunak (CI/CD, reliability, observability), sedangkan MLOps menyoroti siklus hidup ML (data/fitur, eksperimen, model, serving, monitoring). Kotak "titik keterhubungan" menunjukkan praktik yang diterapkan pada proyek ini: otomasi pelatihan lewat pipeline, artefak terversi, deployment berbasis container, dan monitoring produksi.

2.3. Preprocessing Data

Preprocessing dilakukan melalui beberapa tahap untuk menjamin kualitas data sebelum pemodelan. Tahapan meliputi imputasi nilai hilang dengan strategi most_frequent agar distribusi data tetap terjaga, penghapusan duplikasi untuk mencegah bias, konversi tipe data dengan convert_dtypes(), standardisasi fitur numerik menggunakan StandardScaler agar data berada pada skala seragam (mean 0, standar deviasi 1), penghapusan outlier menggunakan metode z-score dengan threshold 3, encoding fitur kategorikal menggunakan LabelEncoder, dan feature engineering berupa binning berbasis kuantil dengan 4 bins guna menangkap pola non-linear pada durasi kunjungan administratif. Penggunaan LabelEncoder dipilih karena seluruh fitur kategorikal bersifat nominal dan jumlah kategorinya relatif terbatas, sehingga encoding sederhana memadai untuk model berbasis pohon yang tidak sensitif terhadap skala jarak antar kode.

Dataset yang telah dipraproses disimpan dalam format CSV untuk tahap pemodelan. Proses preprocessing dijalankan otomatis melalui script Python yang dapat diintegrasikan ke pipeline CI/CD. Strategi ini memastikan ketahanan terhadap imbalance, sebagaimana didukung oleh benchmark literatur pada dataset yang sama.

2.4. Pemilihan dan Pengembangan Model

Random Forest Classifier dipilih karena kemampuannya menangani data berdimensi tinggi, ketahanan terhadap overfitting melalui bagging, kemampuan memodelkan non-linearitas dan interaksi fitur, serta interpretabilitas melalui feature importance [16], [18]. Random Forest merupakan metode ensemble yang menggabungkan banyak decision tree dengan teknik bagging; setiap tree dilatih pada subset hasil bootstrap dan subset fitur acak. Prediksi akhir ditentukan melalui mayoritas voting.

$$\hat{y} = \underset{c}{argmax} \sum_{t=1}^{T} I(h_t(x) = c) \quad (1)$$

Persamaan (1) menyatakan bahwa prediksi akhir (y-hat) ditentukan oleh kelas dengan jumlah prediksi terbanyak dari seluruh decision tree. Setiap decision tree h_t menghasilkan satu prediksi kelas untuk sampel x, lalu mekanisme mayoritas voting memilih kelas c yang paling sering muncul. Pendekatan ini menurunkan variansi dan meningkatkan kemampuan generalisasi dibanding decision tree tunggal.

Dataset dibagi menjadi training 80% dan testing 20% menggunakan train_test_split dengan random_state = 42 agar hasil reprodusibel. Pembagian dilakukan setelah preprocessing untuk mencegah data leakage. Hyperparameter tuning dilakukan menggunakan GridSearchCV dengan validasi silang 3-fold. Hyperparameter yang diuji adalah n_estimators [50, 100, 150] dan max_depth [5, 10, None]. Proses grid search memakai metrik accuracy untuk memilih kombinasi terbaik.

2.5. Evaluasi Model

Evaluasi dilakukan dengan beberapa metrik untuk menggambarkan performa secara komprehensif. Karena dataset tidak seimbang, penilaian tidak hanya mengandalkan accuracy tetapi juga precision, recall, dan F1-score yang lebih representatif untuk kelas minoritas. Rumus metrik evaluasi adalah sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

Akurasi mengukur proporsi prediksi benar terhadap seluruh data uji, namun bisa kurang representatif pada data dengan distribusi kelas tidak seimbang. Karena itu, accuracy digunakan sebagai indikator awal dan perlu dibaca bersama precision, recall, dan F1-score [6].

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \quad (3)$$

Presisi mengukur ketepatan prediksi positif, yaitu proporsi data yang diprediksi positif dan benar-benar positif. Metrik ini penting ketika false positive harus ditekan [7].

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \quad (4)$$

Recall menunjukkan kemampuan model menangkap seluruh data positif yang sebenarnya. Nilai recall tinggi berarti model mampu menemukan sebagian besar kasus positif sehingga penting saat false negative perlu diminimalkan [8].

 ISSN : 2085-9503 (Print), 2581-1355 (On Line)

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

F1-score merupakan harmonic mean antara precision dan recall, berguna pada data tidak seimbang karena menyeimbangkan kesalahan false positive dan false negative. Selain itu, evaluasi juga menggunakan confusion matrix sebagaimana ditunjukkan pada persamaan (6) [9].

$$CM = \begin{matrix} TP & FP \\ FN & TN \end{matrix} \quad (6)$$

Confusion matrix menampilkan distribusi prediksi dan label aktual untuk setiap kelas. Dari matriks ini, metrik seperti accuracy, precision, recall, dan F1-score dihitung secara sistematis. Classification report ditampilkan untuk tiap kelas dan keseluruhan (macro average dan weighted average) agar evaluasi lebih lengkap. Visualisasi confusion matrix menggunakan ConfusionMatrixDisplay, sedangkan feature importance dihitung dan divisualisasikan menggunakan Plotly untuk mengidentifikasi fitur paling berpengaruh [10].

2.6. Implementasi MLOps

Implementasi MLOps memastikan model dapat dikembangkan, di-deploy, dan dimonitor secara efisien. Komponen yang diterapkan meliputi experiment tracking dengan MLflow, di mana setiap run dicatat lengkap dengan parameter, metrik, dan artifacts. MLflow autolog diaktifkan untuk mencatat parameter model, metrik evaluasi, dan artifacts secara otomatis. Model yang telah dilatih disimpan dengan mlflow.sklearn.log_model() guna versioning dan deployment [21].

Untuk MLOps end-to-end, data dan fitur juga di-versioning (mis. Git atau DVC) serta divalidasi melalui pemeriksaan skema (tipe data, nilai hilang, rentang nilai) sebelum training. Snapshot dataset, daftar fitur, dan hash data dicatat bersama run MLflow agar setiap model dapat ditelusuri ke sumber data. Reproducibility diperkuat melalui environment pinning (requirements.txt atau conda), penetapan seed global, dan pencatatan versi library. Model registry digunakan untuk promosi model ke tahap staging dan production melalui approval gate, serta mendukung rollback cepat saat metrik produksi menurun. Token akses dan kredensial disimpan menggunakan GitHub Secrets untuk menjaga keamanan dan audit trail perubahan [21], [25]

Model serving dilakukan dengan MLflow model serving yang menyediakan REST API untuk inference. MLflow serving memudahkan deployment lewat command line atau integrasi dengan containerization tools seperti Docker. Monitoring menggunakan Prometheus dilakukan melalui Prometheus exporter untuk mengumpulkan metrik performa model di produksi. Metrik yang dicatat mencakup inference_duration_seconds untuk waktu inferensi, serta metrik sistem seperti python_gc_collections_total untuk monitoring garbage collection.

Visualisasi melalui Grafana menampilkan dashboard metrik yang dikumpulkan Prometheus secara real-time. Dashboard memuat grafik inference duration, garbage collection metrics, dan metrik lainnya. Alerting rules dikonfigurasi untuk memberi notifikasi ketika metrik melewati threshold. CI/CD pipeline menggunakan GitHub Actions untuk implementasi continuous integration dan deployment [25]. Workflow CI/CD menjalankan training script secara otomatis saat terjadi push ke branch main. Pipeline meliputi checkout repository, setup Python, instalasi dependencies, menjalankan training, dan mengunggah artifacts. Integrasi DagsHub dipakai sebagai remote tracking server untuk MLflow sehingga kolaborasi tim dan akses tracking dapat dilakukan dari berbagai lokasi.

2.7 Alur Eksperimen End-to-End

Untuk memperjelas praktik MLOps, alur eksperimen diringkas sebagai berikut: eksplorasi dan preprocessing dilakukan di Colab/komputer lokal, kode dan konfigurasi disimpan di GitHub untuk versioning, pelatihan dipicu via GitHub Actions (CI) dan setiap run dicatat ke DagsHub/MLflow (parameter, metrik, artefak seperti confusion matrix, kurva ROC/PR, dan model). Model kemudian disajikan melalui MLflow serving (REST API), dikemas dengan Docker, dan dimonitor menggunakan Prometheus serta divisualisasikan/di-alert menggunakan Grafana. Validasi data, pencatatan versi data, registrasi model ke registry, serta monitoring drift yang memicu retraining terjadwal atau berbasis threshold juga dilakukan . Alur ini membentuk feedback loop untuk iterasi model dan menjaga keterulangan (reproducibility) [21], [25].

Gambar 2. Alur Eksperimen MLOps End-to-End

Gambar 2 menekankan urutan aktivitas dan artefak di tiap tahap. Bagian awal (Colab/Local) mendukung iterasi cepat (EDA, preprocessing, baseline). GitHub menjadi source-of-truth kode, sedangkan GitHub Actions menjalankan otomasi (training/evaluasi). DagsHub/MLflow menyimpan riwayat run dan artefak (metrik, model, visualisasi) agar eksperimen dapat dibandingkan. Tahap serving dan Docker memastikan model berjalan konsisten di lingkungan lain, lalu Prometheus/Grafana menyediakan observability (latensi inferensi, error, metrik sistem) sebagai dasar tindakan operasional (alerting dan perbaikan).

2.8. Tools dan Teknologi

Penelitian ini memanfaatkan beragam tools dan teknologi. Python Libraries meliputi pandas untuk manipulasi data, scikit-learn untuk machine learning, numpy untuk komputasi numerik, matplotlib dan plotly untuk visualisasi, serta joblib untuk serialisasi model. MLOps Tools meliputi MLflow untuk experiment tracking dan model serving, Prometheus untuk monitoring, Grafana untuk visualisasi, dan GitHub Actions untuk CI/CD. Development Environment menggunakan Python 3.10, Jupyter Notebook untuk eksplorasi data, dan Git untuk version control. Seluruh dependencies dicatat di requirements.txt agar reproducibility dan setup environment lebih mudah.

3. HASIL DAN PEMBAHASAN

Penelitian ini menyajikan hasil komprehensif dari preprocessing data, hyperparameter tuning, evaluasi Random Forest Classifier, analisis feature importance, serta implementasi MLOps end-to-end untuk deployment produksi. Hasil evaluasi menunjukkan performa yang unggul dibanding baseline dan robust terhadap dataset e-commerce yang tidak seimbang, disertai analisis trade-off bisnis serta implikasi pipeline MLOps.

3.1. Hasil Preprocessing

Preprocessing menghasilkan dataset siap pakai dengan 9.570 sampel setelah outlier dan duplikasi dihapus. Distribusi kelas tetap tidak seimbang: 1.658 sampel kelas positif (pembelian) dan 7.912 sampel kelas negatif (non-pembelian), rasio sekitar 1:4,8. Ketidakseimbangan ini menuntut evaluasi yang cermat karena model cenderung bias ke kelas mayoritas. Fitur numerik sudah distandardisasi (mean mendekati 0, standar deviasi mendekati 1) sehingga semua fitur berada pada skala yang seragam. Fitur kategorikal telah diencode menjadi numerik, memungkinkan pemrosesan oleh algoritma machine learning. Feature engineering menghasilkan fitur Administrative_Duration_Bin yang menangkap pola non-linear pada durasi kunjungan administratif.

Distribusi kelas pascapreprocessing masih imbalance (rasio ~1:4,8 dengan 1.658 pembelian vs 7.912 non-pembelian), kondisi yang lazim pada dataset e-commerce real-time. Meski demikian, Random Forest dinilai cukup robust terhadap imbalance melalui mekanisme bagging, sebagaimana terlihat pada studi sejenis yang melaporkan F1 kelas positif di atas 0,64 tanpa oversampling eksplisit. Dalam penelitian ini, penanganan imbalance tidak diubah secara eksplisit karena fokus utama adalah implementasi MLOps; strategi seperti class_weight, threshold tuning, atau SMOTE direkomendasikan sebagai eksperimen lanjutan yang dapat dijalankan dan dibandingkan melalui MLflow.

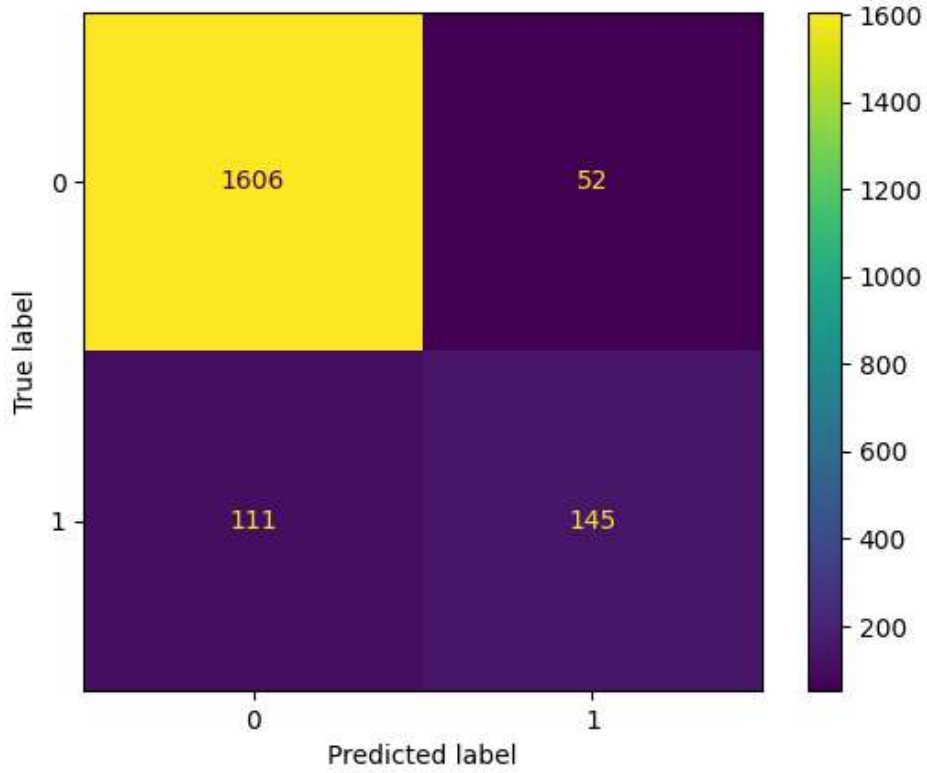
 ISSN : 2085-9503 (Print), 2581-1355 (On Line)

3.2. Hasil Hyperparameter Tuning

GridSearchCV dengan validasi silang 3-fold menghasilkan kombinasi optimal n_estimators=100 dan max_depth=None. Hasil ini menunjukkan manfaat dari trees yang dalam tanpa batas kedalaman maksimum, sementara 100 estimators memberi keseimbangan baik antara performa dan biaya komputasi. Validasi silang memperlihatkan konsistensi performa dengan variasi metrik yang rendah antar fold, menandakan model robust dan tidak overfitting pada data training. Kombinasi terbaik ini kemudian dipakai untuk evaluasi akhir pada testing set.

3.3. Hasil Evaluasi Model

Pada testing set (1.914 sampel: 1.658 non-pembeli dan 256 pembeli), model meraih akurasi 91,38%. Hasil per kelas menunjukkan performa sangat baik dalam mengenali non-pembeli (precision 93,5% dan recall 96,9%). Namun, pada kelas pembeli terdapat trade-off antara precision dan recall. Precision 73,6% menunjukkan bahwa prediksi pembelian benar sekitar 73,6% dari waktu, sedangkan recall 56,6% berarti model hanya menangkap 56,6% pembeli aktual.



Gambar 3. Confusion Matrix Model Random Forest

Gambar 3 menampilkan confusion matrix hasil testing yang menunjukkan distribusi prediksi terhadap label sebenarnya untuk kelas 0 dan 1.

F1-score 64,0% pada kelas pembeli mencerminkan keseimbangan antara precision dan recall. Ketidakseimbangan kinerja kedua kelas berkaitan dengan distribusi dataset yang timpang. Model cenderung konservatif memprediksi kelas positif karena konsekuensi false positive lebih mahal dalam konteks ini. Confusion matrix menunjukkan 1.606 true negatives, 52 false positives, 145 true positives, dan 111 false negatives pada testing set. Jumlah false negative yang lebih tinggi dari false positive mengindikasikan model lebih sering melewatkan pembeli potensial dibanding salah mengklasifikasikan non-pembeli sebagai pembeli. Dalam konteks bisnis yang memprioritaskan biaya false positive, trade-off ini dapat dianggap layak.

Tabel 1. Hasil Classification Report Model Random Forest

Kelas	Precision	Recall	F1-Score	Support
0(Non Pembeli)	0,935	0,969	0,952	1.658
1(Pembeli)	0,736	0,566	0,640	256
Macro Average	0,836	0,768	0,796	1.914
Weighted Average	0,909	0,915	0,910	1.914

Untuk memvalidasi pemilihan Random Forest Classifier, model dibandingkan dengan baseline umum seperti Logistic Regression dan XGBoost pada dataset Online Shoppers Intention yang sama. Perbandingan ini merujuk hasil studi terdahulu (bukan eksperimen ulang pada proyek ini) yang memakai preprocessing serupa (imbalance handling dengan SMOTE atau class_weight), train/test split 80/20, dan metrik standar.

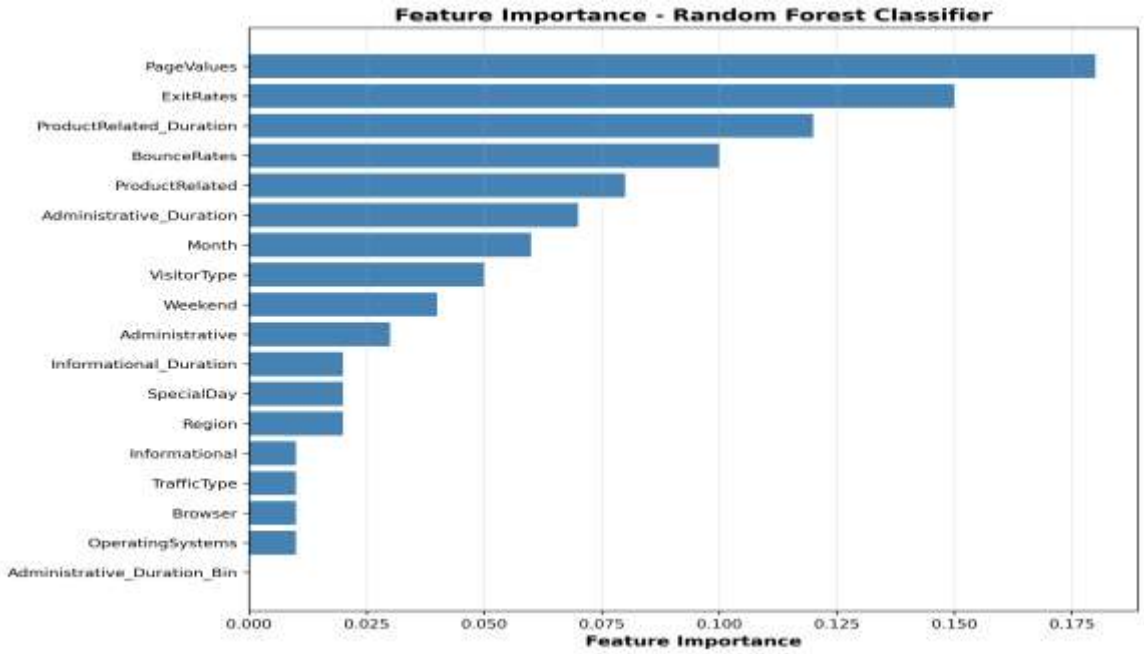
Tabel 2. Perbandingan dengan algoritma lain

Model	Accuracy(%)	Precisiaon(%)	Recall(%)	F1-Score	AUC	Refrensi
Logistic Regression	85,2	65,0	52,0	57,5	0,85	[6]
XGBoost	89,9-93,5	70,0	60,0	64,5	0,92	[7]
Random Forest	91,4	73,6	56,6	64,0	0,90	Penelitian ini

Random Forest konsisten menunjukkan performa kuat pada kelas minoritas (pembelian), dengan akurasi >90% dan F1-score lebih tinggi dibanding Logistic Regression (lemah pada non-linearitas) serta XGBoost (kadang overfit pada dataset tabular kecil). Temuan ini sejalan dengan studi ensemble dan gradient boosting yang menekankan peningkatan prediksi niat pembelian melalui kombinasi model [7], [17]. Tabel perbandingan bersifat rujukan literatur (bukan eksperimen ulang pada proyek ini) dan digunakan untuk menjustifikasi Random Forest sebagai baseline yang stabil dan interpretabel dalam konteks fokus MLOps. Studi lain menggunakan SMOTE untuk menangani imbalance; performa Random Forest kami kompetitif meski tanpa oversampling eksplisit, menegaskan efisiensi ensemble bagging pada dataset e-commerce [19].

3.4. Analisis Feature Importance

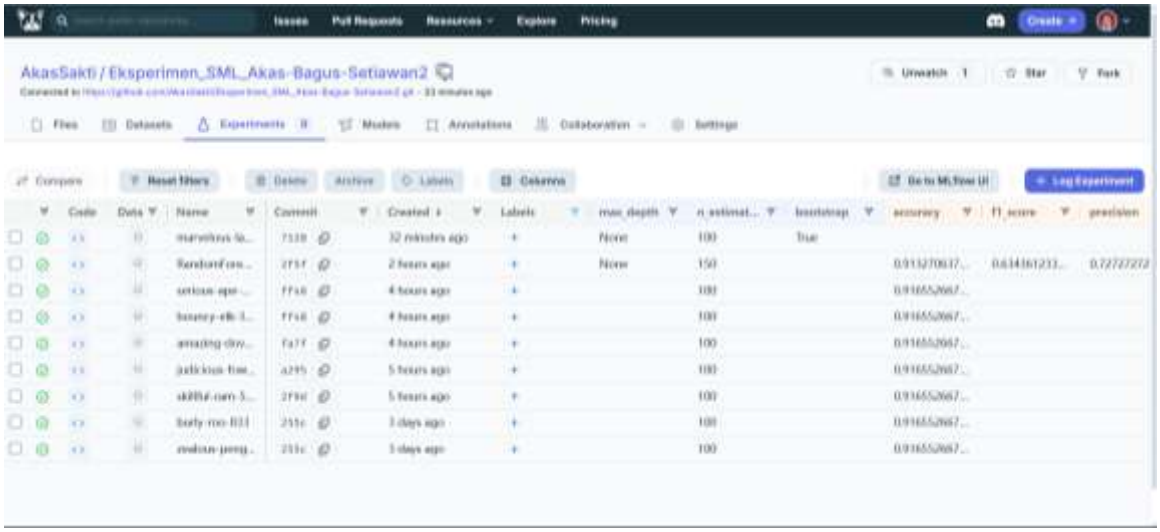
Feature importance menunjukkan bahwa PageValues, ExitRates, ProductRelated_Duration, dan BounceRates merupakan fitur paling berpengaruh. PageValues memiliki importance tertinggi, mengindikasikan nilai halaman yang dikunjungi adalah indikator kuat niat pembelian. Fitur ini merefleksikan nilai ekonomi halaman, di mana halaman bernilai tinggi biasanya terkait produk yang diminati. ExitRates dan BounceRates yang tinggi juga penting, menunjukkan pola navigasi sebagai prediktor signifikan. Pengguna dengan exit rate dan bounce rate rendah cenderung lebih terlibat dan berpeluang membeli lebih tinggi. ProductRelated_Duration memperlihatkan bahwa durasi pada halaman produk adalah indikator minat terhadap produk. Fitur kontekstual seperti Month, VisitorType, dan Weekend memiliki kontribusi moderat, menandakan faktor temporal dan karakter visitor turut memengaruhi niat pembelian meskipun tidak sekuat perilaku browsing.



Gambar 4. Feature Importance Random Forest Classifier.

Gambar 4 menampilkan peringkat feature importance yang memperlihatkan kontribusi relatif tiap fitur terhadap keputusan model.

 ISSN : 2085-9503 (Print), 2581-1355 (On Line)

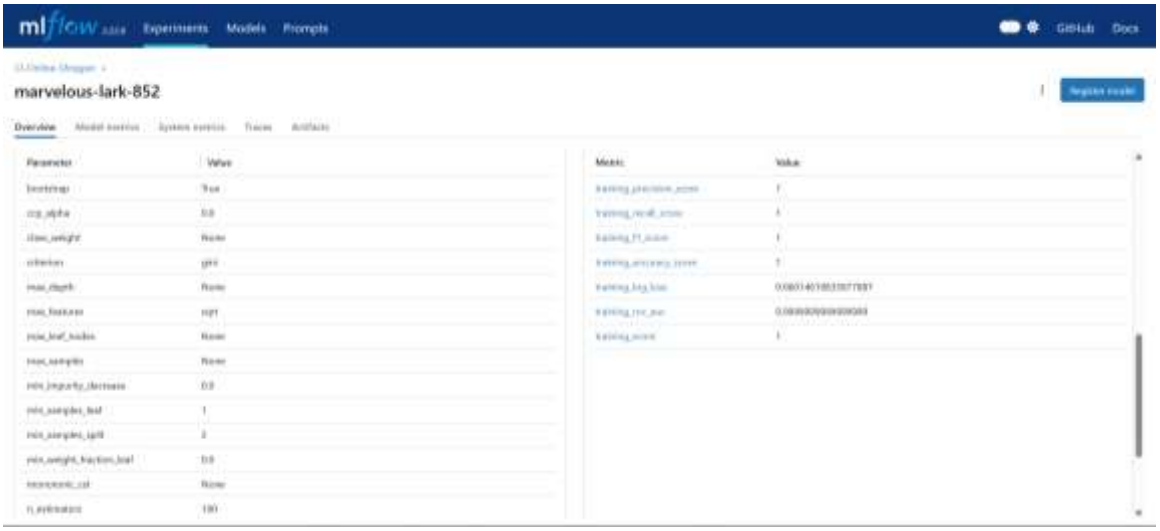


Gambar 5. MLflow Dashboard Experiment Tracking

Gambar 5 menampilkan halaman detail run pada MLflow (DagsHub) yang memuat identitas eksperimen, status run, serta konfigurasi model hasil tuning.

3.5. Implementasi MLOps dan Monitoring

Implementasi MLOps berhasil mengintegrasikan pipeline dari development hingga production. MLflow experiment tracking mencatat semua eksperimen dengan parameter, metrik, artifacts (model, confusion matrix, dan kurva evaluasi), serta versi dependensi yang digunakan [21]. Setiap run dapat dilacak dan dibandingkan melalui MLflow UI, memungkinkan iterasi pengembangan yang efisien. Model serving dengan MLflow berhasil di-deploy dan diakses melalui REST API pada endpoint /invocations. Pengujian menggunakan Postman menunjukkan API berfungsi baik dengan latency yang dapat diterima. CI/CD melalui GitHub Actions menjalankan training otomatis saat ada perubahan kode, sedangkan monitoring dilakukan dengan Prometheus-Grafana untuk memantau durasi inferensi dan metrik sistem.

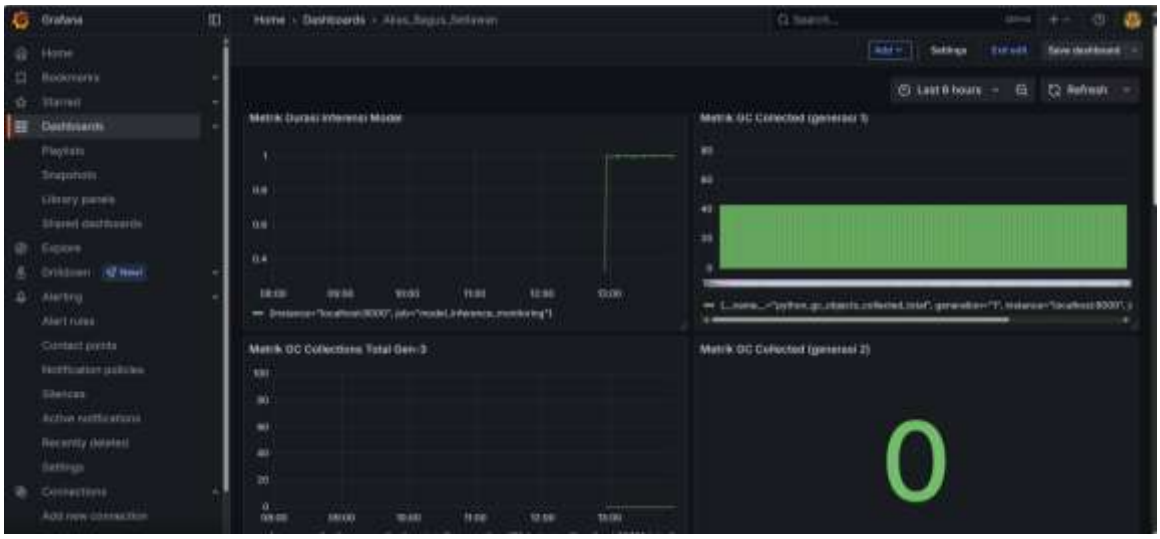


Gambar 6. MLflow Dashboard Overview dengan Experiment Tracking

Gambar 6 menampilkan ringkasan parameter dan metrik utama (accuracy, precision, recall, F1) pada MLflow untuk membandingkan performa antar run.

Prometheus exporter berhasil mengumpulkan metrik inference_duration_seconds dengan rata-rata sekitar 0,1-0,5 detik per inferensi, menandakan performa memadai untuk aplikasi real-time. Metrik garbage collection menunjukkan aplikasi berjalan efisien tanpa memory leak signifikan. Dashboard Grafana menampilkan visualisasi real-time dari berbagai metrik. Alerting rules dikonfigurasi untuk tiga skenario: High Inference Rate ketika jumlah inferensi per detik melampaui threshold, GC Gen 0 Spikes ketika garbage

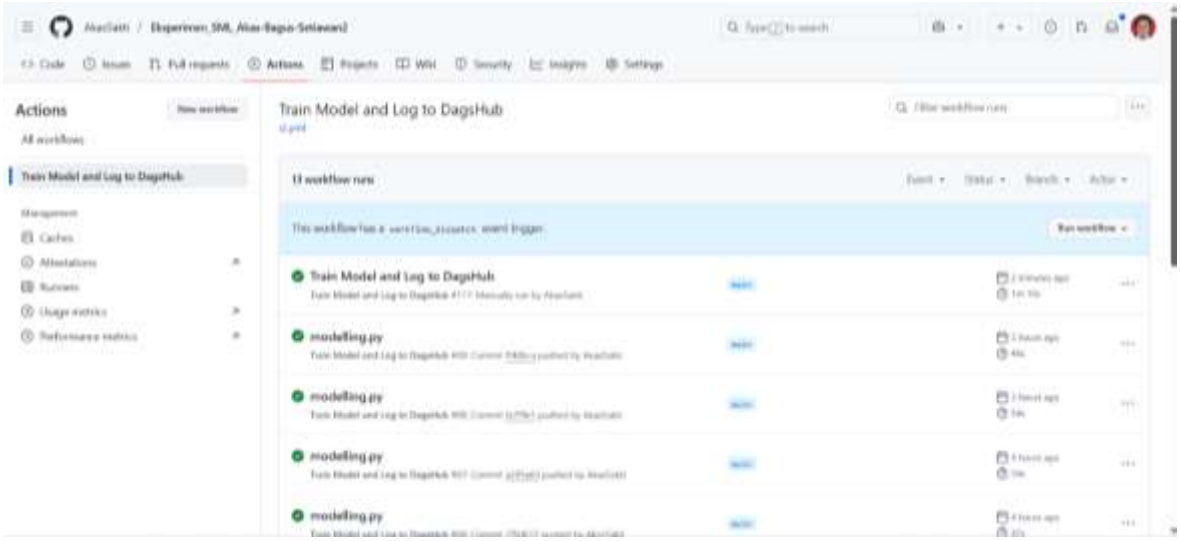
collection generasi 0 meningkat tajam, dan Inference Duration Count untuk memonitor volume inferensi. Alerting berjalan baik dan memberi notifikasi saat kondisi abnormal terdeteksi.



Gambar 7. Grafana Dashboard Monitoring Inference dan Garbage Collection Metrics

Gambar 7 menampilkan dashboard Grafana yang memvisualisasikan metrik durasi inferensi dan metrik garbage collection yang dikumpulkan Prometheus secara real-time.

CI/CD pipeline menggunakan GitHub Actions berhasil mengotomasi proses training dan deployment [25]. Workflow menjalankan training script secara otomatis saat terjadi push ke branch main, memastikan model selalu selaras dengan kode terbaru. Integrasi DagsHub memungkinkan experiment tracking terpusat yang dapat diakses oleh seluruh tim.



Gambar 8. GitHub Actions Workflow untuk CI/CD Pipeline

Gambar 8 menampilkan workflow GitHub Actions untuk pelatihan model dan logging ke DagsHub sebagai bukti CI/CD berjalan otomatis.

3.6. Diskusi dan Implikasi

Hasil penelitian memperlihatkan bahwa Random Forest Classifier efektif untuk memprediksi niat pembelian online shopper dengan akurasi tinggi [16], [18]. Namun, perbedaan performa antar kelas menunjukkan perlunya pertimbangan bisnis. Precision kelas pembeli yang tinggi (73,6%) menandakan prediksi pembelian cukup dapat diandalkan, sehingga cocok untuk personalisasi konten atau targeted marketing. Recall yang lebih rendah (56,6%) mengindikasikan sebagian pembeli potensial terlewat (false negative). Jika biaya kehilangan pembeli dianggap tinggi, strategi yang disarankan adalah menurunkan threshold klasifikasi atau

menerapkan cost-sensitive learning agar false negative berkurang tanpa mengorbankan precision secara drastis. Sebaliknya, jika biaya false positive tinggi, threshold dapat dipertahankan lebih konservatif.

Implementasi MLOps secara menyeluruh memberikan manfaat besar dalam pengembangan dan pemeliharaan model. Experiment tracking memfasilitasi iterasi yang cepat dan pembelajaran dari eksperimen sebelumnya. Monitoring real-time memungkinkan deteksi dini terhadap degradasi performa atau perubahan distribusi data (data drift), sehingga retraining dapat dilakukan proaktif. CI/CD pipeline memastikan perubahan kode dan model bisa di-deploy cepat dan aman, mengurangi waktu menuju production dan menekan risiko human error. Berbagai studi kasus deployment ML menekankan bahwa praktik MLOps komprehensif penting untuk mengatasi kompleksitas operasional [21], [23], [25].

Penelitian ini memiliki keterbatasan. Pertama, dataset mungkin tidak mewakili semua jenis e-commerce atau segmen pasar. Kedua, model hanya menggunakan satu algoritma (Random Forest) sehingga perbandingan dengan XGBoost atau Neural Networks dapat menambah insight. Ketiga, evaluasi dilakukan pada data historis dan validasi pada data real-time production diperlukan untuk mengonfirmasi performa di kondisi nyata. Implikasi praktis meliputi integrasi model untuk personalisasi pengalaman pengguna, penggunaan prediksi untuk optimasi inventory management dan pricing strategy, adopsi MLOps sebagai best practice proyek ML, serta monitoring dan alerting untuk maintenance proaktif model production [18], [21], [25].

4. KESIMPULAN

Penelitian ini mencapai tujuan dengan membangun model prediksi niat pembelian online shopper berbasis Random Forest yang menunjukkan akurasi 91,38%, precision 73,6%, recall 56,6%, dan F1-score 64,0% pada kelas pembeli serta performa sangat baik pada kelas non-pembeli. Implementasi MLOps end-to-end yang mencakup pelacakan eksperimen, deployment, dan monitoring membuktikan bahwa model siap dioperasikan secara andal dan terukur. Temuan ini menegaskan bahwa integrasi Random Forest dengan pipeline MLOps mampu menghasilkan prediksi yang kuat sekaligus kesiapan operasional, sehingga tujuan penelitian tercapai dan memberikan dasar penerapan di konteks e-commerce.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Laboratorium Rekayasa Sistem Informasi Kampus 4 PSDKU Sidoarjo Politeknik Negeri Jember yang menyediakan sumber daya untuk pengembangan proyek ini. Ucapan terima kasih juga disampaikan kepada komunitas open source yang telah mengembangkan tools seperti MLflow, Prometheus, dan Grafana yang memungkinkan implementasi MLOps dalam penelitian ini, dan para pihak yang telah mendukung tersusunnya penelitian ini.

DAFTAR PUSTAKA

[1] M. A. Musababa and M. Fachrie, “Data Streaming Pipeline Model Using DBSTREAM-Based Online Machine Learning for E-Commerce User Segmentation,” Journal of Applied Informatics and Computing, vol. 9, no. 6, pp. 3346–3355, Dec. 2025, doi: 10.30871/jaic.v9i6.11522.

[2] S. Zhang, T. Mo, and Z. Zhang, “LightPersML: A Lightweight Machine Learning Pipeline Architecture for Real-Time Personalization in Resource-Constrained E-commerce Businesses,” Journal of Advanced Computing Systems, vol. 4, no. 8, pp. 44–56, 2024, doi: 10.69987/JACS.2024.40807.

[3] Y. E. Gundogmus, S. Acikalin, and A. Rastak, “Customer Lifetime Value Prediction in E-Commerce: Machine Learning Approaches and Business Implications,” in Proc. Int. Symp. Innovations in Intelligent SysTems and Applications (INISTA), 2025.

[4] O. R. S. G. Alamuri and C. K. Bondalapu, “Predicting E-Commerce Purchase Intention Using Machine Learning,” ASEAN J. Sci. Tech. Report., vol. 29, no. 1, e260159, 2026, doi: 10.55164/ajstr.v29i1.260159.

[5] X. Ma and X. Jiang, “Predicting Cross-border E-commerce Purchase Behavior in Organic Products: A Machine Learning Approach Integrating Cultural Dimensions and Digital Footprints,” International Journal of Computer and Information System, vol. 5, no. 1, pp. 91–102, 2024, doi: 10.29040/ijcis.v5i1.212.

[6] T. Nikhitha, S. S. Sameer, and D. Bhattacharya, “Online Shopping Purchase Intention Prediction Using Machine Learning,” in Proc. Int. Conf. Communication, Computer, and Information Technology (IC3IT), 2025.

[7] A.-A. Tanvir, I. A. Khandokar, A. K. M. M. Islam, S. Islam, and S. Shatabda, "A Gradient Boosting Classifier for Purchase Intention Prediction of Online Shoppers," Heliyon, vol. 9, no. 4, e15163, 2023, doi: 10.1016/j.heliyon.2023.e15163.
[8] T. T. Nguyen, H. T. T. Truong, and T. Le-Anh, "Online Purchase Intention Under the Integration of Theory of Planned Behavior and Technology Acceptance Model," SAGE Open, vol. 13, no. 4, art. 21582440231218814, 2023, doi: 10.1177/21582440231218814.
[9] F. V. Ferdinand, L. Laurence, and K. N. Effendi, "Comparing Random Forest and XGBoost Machine Learning Models for Predicting Purchase Intention in Online Consumer Behavior: A Study in the Jabodetabek Area," in Program & Abstract Book of CIC 2025 (CONMEDIA 2025), Malacca, Malaysia, Oct. 14–17, 2025, p. 94.
[10] A. K. Prasad, D. K. M, V. D. J. Macedo, B. R. Mohan, and A. P. N, "Machine Learning Approach for Prediction of the Online User Intention for a Product Purchase," Int. J. Recent Innov. Trends Comput. Commun., vol. 11, no. 1s, pp. 43–51, 2023, doi: 10.17762/ijritcc.v11i1s.5992.
[11] J. Adhikari, "Online Shoppers' Purchase Intention using Ensemble Learning Approach," International Journal of Next-Generation Computing, vol. 14, no. 4, Nov. 2023, doi: 10.47164/ijngc.v14i4.1065.
[12] C. Clarence and K. Keni, "The Prediction of Purchase Intention Based on Digital Marketing, Customer Engagement, and Brand Preference," in Proc. 10th Int. Conf. Entrepreneurship and Business Management (ICEBM 2021), Adv. Econ., Bus. Manag. Res., Atlantis Press, 2022, pp. 481–486, doi: 10.2991/aebmr.k.220501.073.
[13] R. Kale, K. Bidwai, M. Maske, R. Bansode, and P. Gurav, "Prediction of Customer Purchase Intention using Social Media Data," Int. J. Adv. Res. Sci. Commun. Technol. (IJARSCT), vol. 2, no. 5, pp. 10–13, May 2022, doi: 10.48175/IJARSCT-4003.
[14] Y. Tong, "The Influence of Online Celebrity Live Streaming on Consumers' Purchasing Decisions," Highlights in Business, Economics and Management, vol. 8, pp. 411–418, 2023, doi: 10.54097/hbem.v8i.7239.
[15] S. Sikka and J. Kumar, "Understanding Indian Millennials' Awareness Towards Brand Personality of Apparel Brands," Small Enterprises Development, Management & Extension Journal, vol. 51, no. 1, pp. 41–53, 2024, doi: 10.1177/09708464231209451.
[16] H. A. Salman, A. Kalakech, and A. Steiti, "Random Forest Algorithm Overview," Babylonian Journal of Machine Learning, vol. 2024, pp. 69–79, 2024, doi: 10.58496/BJML/2024/007.
[17] Z. Sun, G. Wang, P. Li, H. Wang, M. Zhang, and X. Liang, "An improved random forest based on the classification accuracy and correlation measurement of decision trees," Expert Systems with Applications, vol. 237, pt. B, art. 121549, 2024, doi: 10.1016/j.eswa.2023.121549.
[18] S. F. Neduet, A. H. Neduet, S. B. Amir, M. G. Z. Awan, S. H. Ahmed, and S. M. H. Aslam, "XGBoost and Random Forest Algorithms: An in Depth Analysis," Pakistan Journal of Scientific Research, vol. 3, no. 1, pp. 26–31, 2023, doi: 10.57041/vol3iss1pp26-31.
[19] H. Hairani, A. Anggrawan, and D. Priyanto, "Improvement Performance of the Random Forest Method on Unbalanced Diabetes Data Classification Using Smote-Tomek Link," Int. J. Informatics Visualization, vol. 7, no. 1, pp. 258–264, 2023, doi: 10.30630/joiv.7.1.1069.
[20] W. Zhang, C. Wu, H. Zhong, Y. Li, and L. Wang, "Prediction of undrained shear strength using extreme gradient boosting and random forest based on Bayesian optimization," Geoscience Frontiers, vol. 12, no. 1, pp. 469–477, 2021, doi: 10.1016/j.gsf.2020.03.007.
[21] D. Kreuzberger, N. Kuhl, and S. Hirschl, "Machine Learning Operations (MLOps): Overview, Definition, and Architecture," IEEE Access, vol. 11, pp. 31866–31879, 2023, doi: 10.1109/ACCESS.2023.3262138.
[22] S. Pahune and Z. Akhtar, "Transitioning from MLOps to LLMOps: Navigating the Unique Challenges of Large Language Models," Information, vol. 16, no. 2, art. 87, 2025, doi: 10.3390/info16020087.
[23] A. Sanchez-Mompo, I. Mavromatis, P. Li, K. Katsaros, and A. Khan, "Green MLOps to Green GenOps: An Empirical Study of Energy Consumption in Discriminative and Generative AI Operations," Information, vol. 16, no. 4, art. 281, 2025, doi: 10.3390/info16040281.
[24] A. C. Cob-Parro, Y. Lalangui, and R. Lazcano, "Fostering Agricultural Transformation through AI: An Open-Source AI Architecture Exploiting the MLOps Paradigm," Agronomy, vol. 14, no. 2, art. 259, 2024, doi: 10.3390/agronomy14020259.
[25] P. Liang, B. Song, X. Zhan, Z. Chen, and J. Yuan, "Automating the training and deployment of models in MLOps by integrating systems with machine learning," Applied and Computational Engineering, vol. 76, pp. 1–7, 2024, doi: 10.54254/2755-2721/76/20240690.