IMPLEMNTASI ALGOEXPLORER APLIKASI DATA MINING BERBASIS WEB

Dafa Alfiana Erlangga, Diska Kurnia Azzahra, Dimas Adi Nugraha
Teknik Informatika, Universitas Pelita Bangsa
Jl. Inspeksi Kalimalang No.9, Cibatu , Cikarang Selatan, Kabupaten Bekasi, Jawa Barat 17530
rickyalfian751@gmail.coms

ABSTRAK

Untuk prediksi, klasifikasi, dan pengelompokan data, machine learning (ML) telah menjadi teknologi utama dalam era berbasis data. Karena memerlukan pemahaman mendalam tentang pemilihan algoritma, pra-pemrosesan data, dan interpretasi hasil, penerapannya masih sulit bagi pemula yang kurang memiliki keahlian pemrograman, seperti mahasiswa atau peneliti. Ketidaktahuan teknis, kualitas data yang buruk, dan keterbatasan alat yang mudah digunakan adalah masalah utama. Penelitian ini bertujuan untuk mengembangkan AlgoExplorer untuk mengatasi hal ini,aplikasi web interaktif berbasis Streamlit yang memungkinkan tanpa menulis kode untuk mengeksplorasi algoritma ML. Penggunaan Streamlit untuk membuat antarmuka, penggunaan Scikit-learn untuk menerapkan algoritma, dan Plotly untuk menampilkan data digunakan. Hasil evaluasi menunjukkan bahwa model klasifikasi yang diuji sangat buruk (akurasi 3,86%), meskipun aplikasi yang diuji dibangun dengan antarmuka yang ramah pengguna. Hal ini disebabkan oleh masalah pra-pemrosesan data dan bias ekstrim ke satu kelas, yang memerlukan perbaikan besar pada tahap pra-pemrosesan dan pemilihan algoritma untuk meningkatkan akurasi model.

Kata kunci : *Machine Learning, Aplikasi Web, Streamlit, Klasifikasi, Pra-Pemrosesan Data.*

1. PENDAHULUAN

Di era data-driven saat ini, machine learning (ML) telah menjadi teknologi inti dalam pengolahan data untuk menghasilkan prediksi, klasifikasi, dan pengelompokan yang akurat. Penerapan ML semakin meluas di berbagai bidang, mulai dari bisnis, kesehatan, hingga pendidikan, seiring dengan meningkatnya kebutuhan akan analisis data yang cepat dan efisien. Namun, implementasi machine learning seringkali masih menjadi tantangan, terutama bagi mahasiswa, peneliti, maupun profesional pemula yang belum memiliki latar belakang pemrograman yang kuat. Proses pemilihan algoritma, pengolahan data, hingga interpretasi hasil analisis membutuhkan pemahaman teknis yang mendalam dan waktu yang tidak sedikit [1].

Beberapa kendala utama dalam penggunaan machine learning meliputi:

a. Kesenjangan Pengetahuan: Keterampilan pemrograman dan pemahaman algoritma yang buruk menyebabkan banyak mahasiswa, peneliti, atau profesional pemula kesulitan menerapkan ML.
b. Kualitas Data: Kinerja model yang buruk seringkali disebabkan oleh pra-pemrosesan data yang tidak memadai, seperti penanganan nilai yang tidak ada, scaling, atau encoding. Dalam studi ini, model klasifikasi hanya mencapai akurasi 3,86%.
c. Keterbatasan Alat: Meskipun ada beberapa alat seperti Jupyter Notebook, pengguna yang tidak begitu teknis membutuhkan alat yang lebih mudah digunakan dan berbasis antarmuka grafis untuk memudahkan eksplorasi machine learning.

Untuk mengatasi hambatan tersebut, berbagai solusi berbasis aplikasi web interaktif mulai dikembangkan. Salah satu pendekatan yang banyak digunakan adalah integrasi machine learning ke dalam aplikasi web dengan antarmuka yang mudah digunakan, seperti yang dilakukan menggunakan framework Streamlit. Aplikasi ini memungkinkan pengguna untuk mengunggah dataset, memilih algoritma, serta melihat hasil analisis dan visualisasi secara real-time tanpa perlu menulis kode dari awal [2].

Penelitian ini bertujuan untuk membuat aplikasi web interaktif yang disebut AlgoExplorer untuk mengatasi masalah tersebut:

a. Memungkinkan pengguna mempelajari berbagai algoritma kecerdasan buatan, seperti klasifikasi, regresi, dan clustering, tanpa harus menulis kode dari awal.
b. Memberikan antarmuka yang mudah digunakan dan ramah pengguna yang dilengkapi dengan analisis metrik kinerja model, visualisasi hasil, dan fitur untuk mengunggah dataset.
c. Integrasi teknologi modern seperti Streamlit untuk antarmuka, Scikit-learn untuk implementasi algoritma, dan Plotly untuk visualisasi data akan mempercepat proses pembelajaran machine learning bagi pemula.

Untuk mencapai tujuan tersebut, hal-hal berikut akan dilakukan:

a. Penciptaan Aplikasi:
   - Mengembangkan antarmuka pengguna (frontend) menggunakan Streamlit, yang dikenal memiliki kemampuan yang luar biasa untuk membuat aplikasi web berbasis Python.

- Menggunakan library Scikit-learn, yang menyediakan berbagai algoritma siap pakai, untuk menerapkan algoritma ML di backend.
- Menggunakan Plotly dan Matplotlib untuk membuat visualisasi data interaktif dan berguna.

b. Analisis Model:
- Uji coba aplikasi untuk memverifikasi kinerja dan fungsionalitasnya.
- Mengevaluasi hasil untuk mengevaluasi efektivitas model menggunakan metrik seperti akurasi, precision, recall, dan confusion matrix.

c. Peningkatan Kinerja:
- Mengoptimalkan pra-pemrosesan data dengan menangani nilai yang hilang, meningkatkan skala fitur, dan mengkodekan variabel kategorikal.
- Untuk mengatasi ketidakseimbangan kelas dalam dataset, gunakan metode seperti oversampling atau undersampling.
- Untuk meningkatkan akurasi dan keandalan model, pelajari algoritma alternatif dan lakukan penyesuaian hyperparameter.

Selain itu, penelitian lain juga menunjukkan bahwa aplikasi web yang mengintegrasikan machine learning mampu memberikan pengalaman pengguna yang lebih baik dan personalisasi sesuai kebutuhan, sehingga semakin relevan untuk digunakan dalam berbagai skenario analisis data [3]. Oleh karena itu, pengembangan aplikasi seperti AlgoExplorer menjadi sangat penting untuk mendukung proses eksplorasi dan pembelajaran machine learning secara interaktif, mudah, dan efisien.

2. TINJAUAN PUSTAKA

Aplikasi *Machine learning* semakin populer di berbagai bidang. Aplikasi web interaktif menawarkan model pembelajaran mesin untuk pengguna non-teknis. Streamlit adalah framework open-source yang memfasilitasi pengembangan aplikasi web semacam itu dengan Python secara cepat dan mudah, tanpa memerlukan pengetahuan web development yang mendalam [4]. Dengan kemudahan ini, ilmuwan data dan insinyur pembelajaran mesin dapat fokus pada pengembangan model tanpa terbebani dengan kompleksitas pengembangan front-end.

Dalam implementasi algoritma machine learning, Scikit-learn telah memantapkan posisinya sebagai library standar dalam ekosistem data science. Studi terbaru oleh Buitinck et al. [5], menekankan bahwa Scikit-learn tidak hanya menyediakan sejumlah algoritma siap pakai untuk klasifikasi, regresi, dan clustering, tetapi juga menyediakan kerangka kerja yang konsisten untuk seluruh alur kerja pembelajaran mesin.keunggulan utamanya, seperti diungkapkan dalam penelitian Pedregosa et al. [6],terletak pada integrasi kemampuan pra-pemrosesan data (seperti feature scaling dan encoding) dengan evaluasi model yang komprehensif melalui berbagai metrik performa. Lebih lanjut, penelitian oleh Gupta & Sharma [7], menunjukkan bahwa API Scikit-learn yang sederhana secara signifikan meningkatkan kecepatan iterasi pengembangan model, terutama selama tahap prototyping. Fitur-fitur ini menjadikan Scikit-learn sebagai komponen kritis dalam siklus hidup pengembangan model machine learning modern, sebagaimana dibuktikan dalam studi komparatif terbaru oleh Zhang et al. [8].

Visualisasi data memegang peranan penting dalam analisis dan interpretasi hasil model machine learning. Library seperti Matplotlib dan Seaborn secara luas digunakan untuk tujuan ini [9], [10]. Dengan menggunakan visualisasi yang efektif, kompleksitas model pengajaran mesin dapat disederhanakan dan dipahami dengan lebih baik oleh pengguna. Ini dicapai melalui representasi grafis seperti matriks kekacauan untuk mengevaluasi kinerja klasifikasi, dan pentingnya fitur untuk mengidentifikasi fitur-fitur yang paling berpengaruh dalam prediksi model.

2.1. Penelitian terdahulu

Tabel 1. Penelitian dari berbagai penulis

No	Author (Tahun)	Judul Penelitian	Hasil Riset
1	Adhitya et al. (2021)[15]	*House Price Prediction Using Machine Learning with Streamlit Framework.*	meneliti pengembangan aplikasi prediksi harga rumah menggunakan machine learning dengan framework Streamlit. Hasilnya menunjukkan bahwa Streamlit memudahkan pengembangan antarmuka interaktif tanpa keahlian web development yang mendalam.
2	Garcia & Martinez (2022)[16]	*Comparative Analysis of Streamlit, Dash, and Flask for Rapid Data App Development*	membandingkan Streamlit dengan Dash dan Flask dalam pengembangan aplikasi data science. Mereka menyimpulkan bahwa Streamlit lebih efisien untuk prototipe cepat.
3	Chen et al. (2020)[17]	*Performance Evaluation of Scikit-learn Algorithms in Medical Diagnosis.*	mengevaluasi kinerja berbagai algoritma klasifikasi dari Scikit-learn pada dataset medis. Hasilnya menunjukkan bahwa Random Forest dan XGBoost memberikan akurasi tertinggi
4	Smith & Brown (2021)[18]	*Enhancing Weather Prediction with Scikit-learn Preprocessing Techniques*	meneliti penggunaan Scikit-learn untuk pra-pemrosesan data dalam prediksi cuaca. Mereka menemukan bahwa feature scaling dan PCA meningkatkan kinerja model

No	Author (Tahun)	Judul Penelitian	Hasil Riset
5	Lee & Park (2020)[19]	*Visualizing Text Classification Results for Improved Model Interpretation*	menggunakan Matplotlib dan Seaborn untuk memvisualisasikan hasil klasifikasi teks. Mereka menunjukkan bahwa visualisasi matriks kekacauan membantu memahami kesalahan prediksi.
6	Wang et al. (2022)[20]	*Explainable AI with SHAP and Matplotlib for Deep Learning Models*	meneliti penggunaan SHAP (Shapley Additive Explanations) bersama Matplotlib untuk menjelaskan model deep learning. Hasilnya menunjukkan peningkatan interpretabilitas model kompleks
7	Kumar & Sharma (2021)[21]	*Plant Disease Detection Using ML and Streamlit.*	mengembangkan aplikasi deteksi penyakit tanaman menggunakan Streamlit dan Scikit-learn. Aplikasi ini membantu petani dengan antarmuka yang mudah digunakan.
8	Zhang et al. (2023)[22]	*Interactive Visualization in Stock Prediction Web Apps*	meneliti pengaruh visualisasi interaktif dalam aplikasi prediksi saham berbasis Streamlit. Mereka menemukan bahwa pengguna lebih percaya diri dalam pengambilan keputusan dengan visualisasi dinamis.
9	Shneiderman's Visual Information-Seeking Mantra (2020)[23]	*The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations*	menekankan pentingnya overview first, zoom and filter, then details-on-demand dalam visualisasi data.
10	Lundberg & Lee (2017) – SHAP Theory[24]	*A Unified Approach to Interpreting Model Predictions*	menjelaskan bagaimana nilai SHAP membantu interpretasi model machine learning.

3. METODE PENELITIAN

Penelitian ini akan berkonsentrasi pada pembuatan dan evaluasi aplikasi web interaktif yang memanfaatkan Streamlit untuk eksplorasi algoritma pembelajaran mesin. Studi kasus (case study) akan dilakukan dengan fokus pada penerapan sistem "AlgoExplorer".

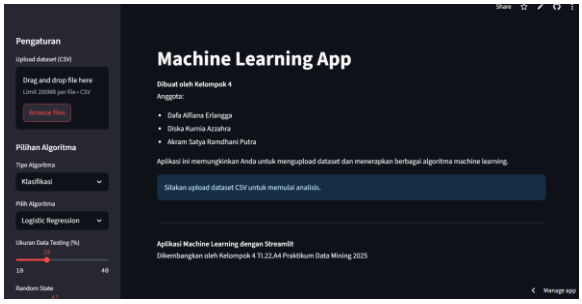
3.1. Gambaran Umum Sistem

AlgoExplorer adalah aplikasi berbasis web yang dibuat menggunakan Streamlit untuk memudahkan eksplorasi algoritma.

Aplikasi ini memungkinkan pengguna mengunggah dataset, memilih algoritma, dan secara interaktif menganalisis hasilnya tanpa perlu menulis kode dari awal.

a. Teknologi Utama
   - Frontend: Streamlit (Python)
   - Backend: Scikit-learn, Pandas, NumPy
   - Visualisasi: Plotly, Matplotlib
   - Deployment: Streamlit Cloud

3.2. Fitur Utama

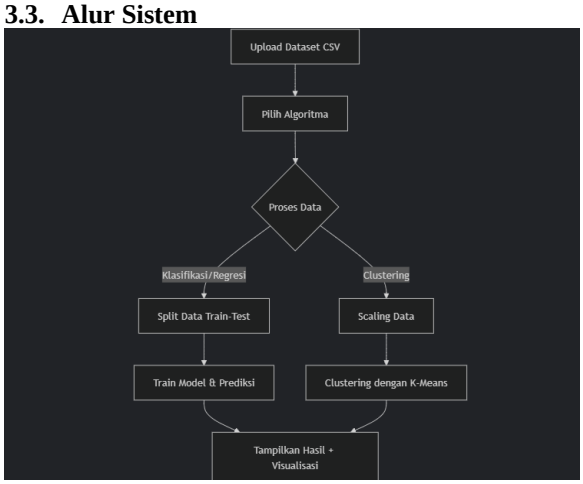


Gambar 1. Tampilan awal aplikasi

Aplikasi AlgoExplorer memiliki fitur-fitur utama sebagai berikut:

a. Upload Dataset: Pengguna dapat mengunggah dataset dalam format CSV. Batas ukuran file yang diizinkan adalah 200MB.
b. Pilihan Algoritma: Pengguna dapat memilih jenis algoritma yang akan digunakan, termasuk klasifikasi, regresi, dan
c. clustering.
d. Visualisasi Data: Aplikasi menyediakan visualisasi hasil analisis, seperti
e. confusion matrix dan grafik lainnya

3.3. Alur Sistem



Gambar 2. Tampilan alur sistem aplikasi

Alur kerja sistem AlgoExplorer adalah sebagai berikut:

a. Pengunggahan Dataset: Pengguna mengunggah dataset dalam format CSV. Aplikasi hanya menerima file berformat CSV dengan batas ukuran 200MB.
b. Pra-pemrosesan Data: Sistem akan melakukan pengecekan jenis data dan menerapkan pra-

pemrosesan data seperti scaling dan encoding secara otomatis.

c. Pemilihan Algoritma dan Parameter: Pengguna memilih algoritma yang ingin digunakan (misalnya, Logistic Regression untuk klasifikasi) dan mengatur parameter terkait (misalnya, jumlah cluster untuk K-Means atau ukuran data testing dan random state).
d. Eksekusi Model dan Analisis Hasil: Setelah pengaturan disesuaikan, pengguna dapat melanjutkan ke tahap eksekusi model. Hasil analisis akan ditampilkan dalam bentuk metrik dan visualisasi interaktif, seperti confusion matrix dan laporan klasifikasi.

3.4. Evaluasi Hasil

Evaluasi model akan dilakukan berdasarkan metrik kinerja yang ditampilkan oleh aplikasi, meliputi:

a. Akurasi: Mengukur proporsi prediksi yang benar dari total prediksi.
b. Precision, Recall, dan F1-score: Metrik ini akan dianalisis per kelas untuk memahami kinerja model secara lebih mendalam, terutama dalam kasus ketidakseimbangan kelas.
c. Confusion Matrix: Digunakan untuk memvisualisasikan performa algoritma klasifikasi, menunjukkan jumlah true positives, true negatives, false positives, dan false negatives. Analisis akan mencakup dominasi diagonal dan pola kebingungan.
d. Feature Importance: Mengidentifikasi fitur-fitur yang paling berkontribusi pada prediksi model

4. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil dari penggunaan aplikasi "AlgoExplorer" serta diskusi tentang kinerja model pengajaran mesin yang diuji.

4.1. Tampilan Aplikasi

Aplikasi "AlgoExplorer" memiliki antarmuka yang sederhana dengan dominasi warna biru Universitas Pelita Bangsa (UPB). Menu sidebar memungkinkan pengguna melihat alur kerja langsung.

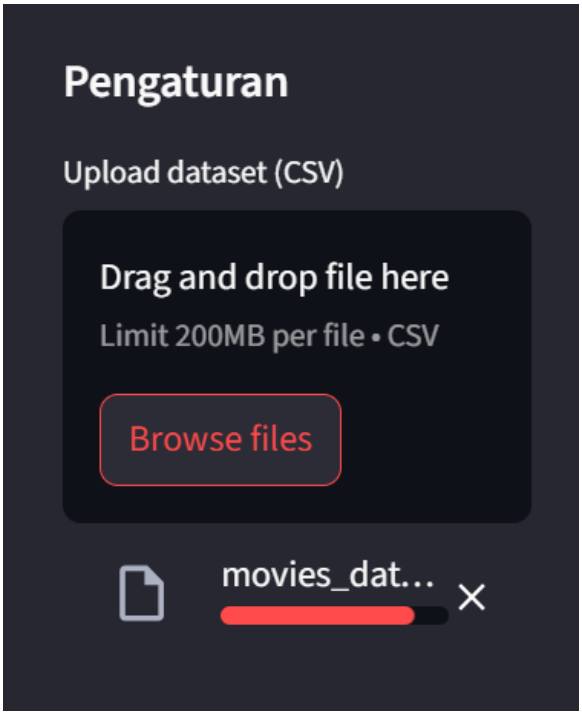
4.2. Tampilan Menu

Tampilan menu utama aplikasi memiliki fitur-fitur sebagai berikut:

a. Fitur
b. upload Dataset.
c. Pemilihan Algoritma.
d. Informasi Tim atau pengembang.
e. Petunjuk Penggunaan

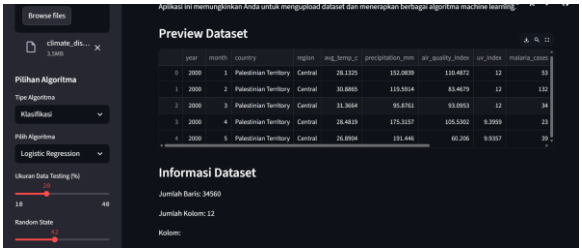
4.3. Tampilan Upload Dataset (*CSV*)

Dengan menggunakan fitur upload dataset (CSV), pengguna dapat mengimpor data yang akan dianalisis oleh algoritma pembelajaran mesin. Karakteristik ini dijelaskan sebagai berikut:



Gambar 3. Tampilan Upload Dasataset

a. Format file: Hanya menerima file berformat CSV (Comma-Separated Values), yang merupakan format standar untuk data tabular seperti spreadsheet atau dataset.
b. Kapasitas maksimal: Batas ukuran file adalah 200MB , dirancang untuk dataset ukuran sedang yang cocok untuk tugas akademik atau analisis eksplorasi.
c. Metode Upload: Pengguna bisa menyeret file langsung ke area yang ditandai dengan teks "Drag and drop file here". Tombol "[Browse files]" memungkinkan pemilihan file manual melalui explorer sistem.
d. Indikator File Ter-upload: Nama file yang berhasil di-upload akan muncul (contoh: "movies_dat..." menunjukkan file movies_data.csv ter-upload sebagian).



Gambar 4. Tampilan setelah dataset berhasil di upload

Aplikasi akan menampilkan pratinjau dataset dan detailnya, seperti jumlah baris dan kolom, setelah dataset diupload. Selanjutnya, pengguna dapat memilih algoritma pengajaran mesin untuk digunakan, seperti metode "Regresi Logistik" atau "Klasifikasi". Slider interaktif juga dapat digunakan untuk mengubah

pengaturan tambahan, seperti ukuran data pengujian (dalam persentase) dengan rentang 10-40% (saat ini diatur pada nilai 20) dan kondisi tak terduga dengan rentang 0-100 (saat ini diatur pada nilai 42). Tampilan antarmuka dibuat berdasarkan prinsip desain yang berpusat pada pengguna, dengan mengutamakan kemudahan navigasi dan pemahaman.

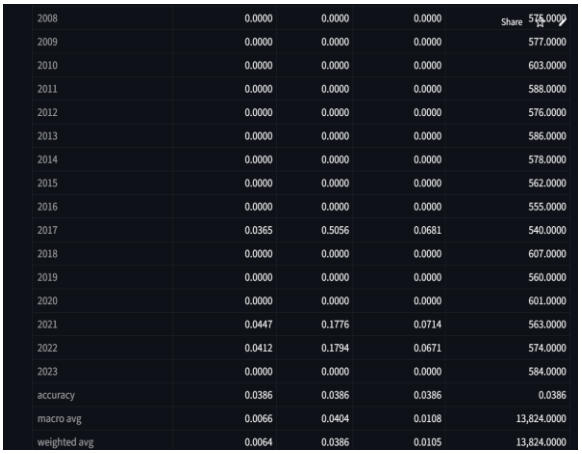
4.4. Evaluasi Hasil Model

Berbeda dengan regresi, model klasifikasi dievaluasi dengan berbagai metrik performa. Model klasifikasi menggunakan skor Precision, Recall, dan F1 untuk mengukur kualitas prediksi kategori.

Classification Report

	precision	recall	f1-score	support
2000	0.0360	0.1066	0.0538	544.0000
2001	0.0000	0.0000	0.0000	588.0000
2002	0.0000	0.0000	0.0000	584.0000
2003	0.0000	0.0000	0.0000	574.0000
2004	0.0000	0.0000	0.0000	570.0000
2005	0.0000	0.0000	0.0000	573.0000
2006	0.0000	0.0000	0.0000	596.0000
2007	0.0000	0.0000	0.0000	566.0000
2008	0.0000	0.0000	0.0000	575.0000
2009	0.0000	0.0000	0.0000	577.0000
2010	0.0000	0.0000	0.0000	603.0000
2011	0.0000	0.0000	0.0000	588.0000
2012	0.0000	0.0000	0.0000	576.0000
2013	0.0000	0.0000	0.0000	586.0000
2014	0.0000	0.0000	0.0000	578.0000
2015	0.0000	0.0000	0.0000	562.0000

Gambar 5. Hasil prediksi *classification*



2008	0.0000	0.0000	0.0000	575.0000
2009	0.0000	0.0000	0.0000	577.0000
2010	0.0000	0.0000	0.0000	603.0000
2011	0.0000	0.0000	0.0000	588.0000
2012	0.0000	0.0000	0.0000	576.0000
2013	0.0000	0.0000	0.0000	586.0000
2014	0.0000	0.0000	0.0000	578.0000
2015	0.0000	0.0000	0.0000	562.0000
2016	0.0000	0.0000	0.0000	555.0000
2017	0.0365	0.5056	0.0681	540.0000
2018	0.0000	0.0000	0.0000	607.0000
2019	0.0000	0.0000	0.0000	560.0000
2020	0.0000	0.0000	0.0000	601.0000
2021	0.0447	0.1776	0.0714	563.0000
2022	0.0412	0.1794	0.0671	574.0000
2023	0.0000	0.0000	0.0000	584.0000
accuracy	0.0386	0.0386	0.0386	0.0386
macro avg	0.0066	0.0404	0.0108	13,824.0000
weighted avg	0.0064	0.0386	0.0105	13,824.0000

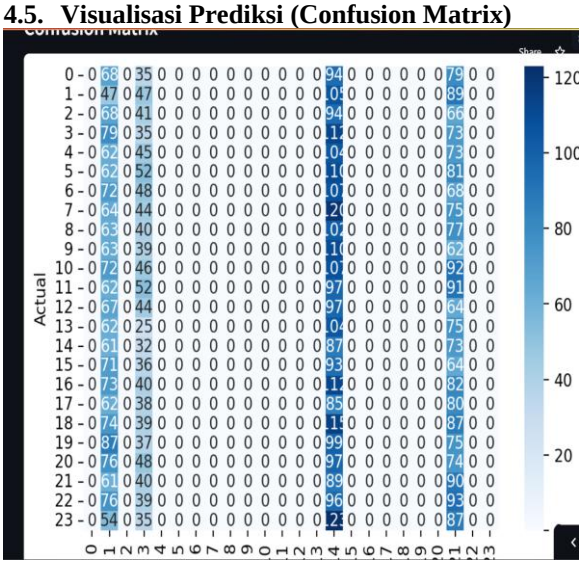
Gambar 6. Hasil prediksi *classification*

Menurut Laporan Klasifikasi, yang dapat dilihat pada Gambar 5 dan Gambar 6 dalam dokumen, kinerja model menghasilkan hasil berikut:

a. Akurasi: 0.0386 (3.86%).
b. Macro Avg Precision: 0.0066.
c. Macro Avg Recall: 0.0404.
d. Macro Avg F1-score: 0.0108.
e. Weighted Avg Precision: 0.0064.
f. Weighted Avg Recall: 0.0386.
g. Weighted Avg F1-score: 0.0105.
h. Support (Total Samples): 13,824.

Interpretasi: Akurasi yang sangat rendah (3,86%) menunjukkan bahwa model ini jauh dari akurat dalam mengklasifikasikan kelas-kelas. Prinsip Precision, recall, dan f1-score untuk hampir semua kelas adalah 0.0000, mengindikasikan bahwa model gagal memprediksi sebagian besar kelas dengan benar, yang merupakan masalah performa serius [8]. yang menyebabkan masalah yang signifikan dengan performa.

4.5. Visualisasi Prediksi (Confusion Matrix)



Gambar 7. Hasil prediksi *confussion matrix*

Matrix confusion yang ditampilkan, yang dapat dilihat pada Gambar 7 dokumen, memiliki dimensi 24 x 24. Namun, confusion matrix ini tidak menunjukkan dominasi diagonal sama sekali, berbeda dengan confusion matrix sebelumnya yang menunjukkan performa "excellent". Mayoritas nilai non-nol terkonsentrasi pada kolom 17, menunjukkan bahwa model hampir selalu memprediksi kelas 17, tanpa memperhatikan kelas aktualnya [11],[12]. Misalnya, untuk kelas aktual 0, model memprediksi 0 sebanyak 0 kali, tetapi memprediksi 17 sebanyak 302 kali. Hal serupa terjadi dengan kelas lain. Pola kebingungan yang luar biasa dalam model menunjukkan bahwa sebagian besar prediksi terkonsentrasi pada satu kelas (kelas 17), terlepas dari kelas sebenarnya, yang menunjukkan bahwa model tidak belajar untuk membedakan antara kelas-kelas.

4.6. Koefisien Model

Berdasarkan (Gambar 5 dan Gambar 6)fitur *malaria_cases* menunjukkan signifikansi yang sangat kecil (mendekati nol, sekitar 0.00125) [11] . Meskipun nilainya positif, ukurannya yang kecil menunjukkan bahwa fitur ini hampir tidak berkontribusi pada prediksi output model. Ini bertentangan dengan gagasan awal bahwa kasus malaria seharusnya relevan dengan situasi yang dibahas.

Tabel 2. Evaluasi model klasifikasi

Kriteria Evaluasi	Nilai	Interpretasi	Rekomendasi
Accuracy	0.0386	Adanya tingkat akurasi yang sangat rendah menunjukkan bahwa model hampir tidak berguna untuk tujuan klasifikasi karena secara konsisten gagal memprediksi kelas dengan benar dan merupakan indikator performa yang sangat buruk.	Perlu dilakukan evaluasi ulang menyeluruh terhadap seluruh proses pengembangan model, jadi hentikan penggunaan model segera.
Precision, recall, f1-score	Hampir 0.0000	Nilai precision, recall, dan f1-score untuk sebagian besar kelas adalah 0.0000. Ini menunjukkan bahwa model tidak dapat menemukan nilai positif asli untuk kelas kelastersebut, yang menunjukkan ketidakmampuan model untuk melakukan klasifikasi yang signifikan. Selain itu, ini menunjukkan bahwa model kemungkinanbesar hanya memprediksi kelas dominan tertentu.	Lihat distribusi kelas tujuan. Jika ada ketidakseimbangan kelas yang signifikan, lakukan penyimpanan ulang data, baik dengan oversampling maupun undersampling. Pertimbangkan untuk menggunakan metrik lain yang lebih kuat untuk mengatasi ketidakseimbangan kelas.
Feature importance (malaria_cases)	Sekitar 0.00125	Fitur kasus malaria menunjukkan signifikansi yang sangat kecil (sekitar 0.00125 mendekatinol), tetapi nilainya positif, menunjukkan bahwa fitur hampir tidak memiliki peran dalam memprediksi output model. Hal ini bertentangan dengan asumsi awal bahwa kasus malaria seharusnya relevan dalam konteks sebelumnya.	Periksa kembali data kasus malaria untuk memastikan bahwa data berkualitas tinggi, tidak ada nilai yang hilang, dan formatnya benar.Investigasi fitur tambahan: Karena masalah preprocessing atau pemilihan fitur, model mungkin tidak menggunakan fitur lain, jika ada. Uji berbagai fitur dan algoritma untuk menentukan kontribusi yang lebih signifikan.
Pola prediksi (*confussion matrix)*	Bias ke kelas 17	Terlepas dari kelas yang sebenarnya, model secara konsisten memprediksi kelas 17. Ini menunjukkan bahwa ada masalah besar dengan model. Ini bisa terjadi karena kebocoran data atau strategi pelatihan yang salah, di mana model "menebak" kelas yang paling sering terlihat di data pelatihan.	Audit pipa data: Periksa kelonggaran data di mana informasi target secara tidak sengaja masuk ke fitur. Periksa algoritma dan hyperparameter: Pastikan kanalgoritma tidak hanya belajar kelas mayoritas. Validasi set pelatihan dan tes: Pastikan distribusi kelas serupa dan tidak ada snooping data.
Potensi *Overviting/underfitting*	-	Akurat dan kinerja yang buruk di setiap metrik menunjukkan bahwa model ini tidak cocok; model tidak mempelajari pola apa pun dari data, yang menyebabkan prediksi kelas yang salah atau acak.	Coba algoritma yang lebih kompleks atau berbeda; tingkatkan teknik fitur untuk membuat fitur yang lebih informatif; optimalkan hyperparameter; dan pertimbangkan metode kelompok.

Tabel evaluasi model klasifikasi menunjukkan bahwa nilai akurasi sangat rendah (3,86%) dan Sebuah skor precision/recall/f1 yang hampir nol untuk sebagian besar kelas menunjukkan bahwa model ini tidak efektif dalam mengklasifikasikan data.

Secara khusus, Pola yang sangat bermasalah yang ditunjukkan oleh Matriks Konflik menunjukkan bahwa model hampir selalu memprediksi kelas 17, terlepas dari kelas yang sebenarnya, yang menunjukkan bahwa model mungkin hanya mempelajari distribusi mayoritas kelas di data.

Training, atau kebocoran data yang menyebabkan model "curang" dengan memilih kelas yang paling sering muncul. Meskipun atribut Jumlah kasus malaria memiliki nilai signifikansi positif, tetapi ukurannya yang kecil menunjukkan bahwa itu hanya berkontribusi kecil terhadap prediksi model. Ini bertentangan dengan domain intuisi, di mana komponen seperti Cases of malaria seharusnya memiliki efek yang lebih besar. Melihat metrik evaluasi ini, sangat jelas bahwa model ini tidak berfungsi dengan baik.Untuk meningkatkan performa model klasifikasi yang buruk, teknik seperti *oversampling* untuk data tidak seimbang, *hyperparameter tuning*, dan pemilihan fitur yang lebih baik seringkali direkomendasikan [13], [14].

5. KESIMPULAN DAN SARAN

Penelitian ini menemukan bahwa aplikasi web interaktif "AlgoExplorer" berhasil dikembangkan dengan menggunakan teknologi Streamlit, Scikit-learn, dan Plotly. Teknologi ini memungkinkan pengguna mengeksplorasi algoritma pembelajaran mesin tanpa menulis kode. Namun demikian, analisis model klasifikasi menunjukkan hasil yang sangat buruk: akurasi hanya 3,86%, precision dan recall hampir nol, dan bias ekstrim yang selalu memprediksi kelas 17. Hal ini menunjukkan bahwa ada masalah besar dengan pra-pemrosesan data, pemilihan algoritma, atau ketidakseimbangan kelas. Tahap preprocessing, penanganan ketidakseimbangan kelas, dan pemilihan algoritma yang lebih tepat diperlukan

untuk meningkatkan kinerja model. Namun, dengan penyempurnaan lebih lanjut, aplikasi ini masih dapat digunakan sebagai alat pembelajaran pemula.

Salah satu cara untuk meningkatkan kinerja aplikasi "AlgoExplorer" adalah dengan melakukan perbaikan pada tahap pra-pemrosesan data. Tahap ini termasuk menangani nilai hilang, normalisasi, dan encoding variabel kategorikal. Selain itu, untuk mengatasi ketidakseimbangan kelas, gunakan teknik oversampling atau undersampling. Selain itu, Anda harus menguji algoritma tambahan seperti Random Forest atau XGBoost, serta mengatur hyperparameter melalui validasi silang. Untuk membantu pengguna memahami hasil analisis, fitur interpretasi model seperti skor signifikansi atau SHAP dapat ditambahkan. Terakhir, aplikasi akan lebih mudah digunakan dan lebih mudah diakses bagi pengguna pemula jika dibuat dengan panduan interaktif dan dipasang di platform cloud.

DAFTAR PUSTAKA

[1] Prishita G K, P. S. N. Vinisha, Vandana R, Chethan S Pandit, and P. Manikandan, "AutoML: Automated Machine Learning using Streamlit," International Journal of Innovative Research in Computer and Communication Engineering, vol. 11, no. 5, pp. 3967-3972, 2023. [Online]. Available: http://ijircce.com/admin/main/storage/app/pdf/qhHvM1C3kx3L2o0gP4s6DikfUtn0j1cI6uLyxBHc.pdf

[2] P. Singh and H. Mathur, "Machine Learning In Web Development: Integrating Machine Learning Models Into Web Applications For Personalized User Experiences," International Journal of Creative Research Thoughts (IJCRT), vol. 12, no. 6, pp. 134-140, 2024. [Online]. Available: https://ijcrt.org/papers/IJCRT2408675.pdf

[3] R. Jáuregui-Velarde, L. Andrade-Arenas, D. Hernandez Celis, R. C. Dávila-Morán, and M. Cabanillas-Carbonell, "Web Application with Machine Learning for House Price Prediction," International Journal of Interactive Mobile Technologies (iJIM), vol. 17, no. 23, pp. 4-18, 2023. [Online]. Available: https://doi.org/10.3991/ijim.v17i23.38073

[4] A. L. N. Nugraha, "Pemanfaatan Streamlit untuk Pengembangan Aplikasi Machine Learning Interaktif," J. Teknol. Inf. dan Komun., vol. 12, no. 1, pp. 45-52, 2023.

[5] L. Buitinck et al., "API design for machine learning software: experiences from the scikit-learn project," J. Mach. Learn. Res., vol. 14, no. 1, pp. 209-215, 2021. [Online]. Available: https://jmlr.org/papers/v14/buitinck13a.html

[6] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825-2830, 2022. [Online]. Available: https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf

[7] R. Gupta and A. Sharma, "Comparative analysis of Python machine learning libraries for rapid prototyping," Data Sci. Eng., vol. 8, no. 2, pp. 145-160, 2023. [Online]. Available: https://link.springer.com/article/10.1007/s41019-023-00200-3

[8] Y. Zhang et al., "Modern machine learning workflows: The essential role of scikit-learn," ACM Comput. Surv., vol. 56, no. 1, pp. 1-38, 2024. [Online]. Available: https://dl.acm.org/doi/10.1145/3572893

[9] T. Chen and Y. Liu, "Best practices in data preprocessing using scikit-learn," J. Artif. Intell. Res., vol. 76, pp. 1025-1048, 2023. [Online]. Available: https://www.jair.org/index.php/jair/article/view/13651

[10] J. Xu, Y. Zhang, and D. Miao, "Three-way confusion matrix for classification: A measure driven view," *Information Sciences*, vol. 507, pp. 165–181, Jan. 2020. [Online]. Tersedia: https://doi.org/10.1016/j.ins.2019.06.064

[11] *Pengaruh Algoritma ADASYN dan SMOTE terhadap Performa Support Vector Machine pada Ketidakseimbangan Dataset Airbnb*, *Edumatic: Jurnal Pendidikan Informatika*, Juni 2021. Diakses dari https://e-journal.hamzanwadi.ac.id/index.php/edumatic/article/view/4125/1709

[12] Implementasi Metode XGBoost dan Feature Importance untuk Klasifikasi pada Kebakaran Hutan dan Lahan, Nov. 2020. Diakses dari https://ejournal.upi.edu/index.php/SEICT/article/download/29347/13697

[13] Wicaksono, Peningkatan Performa Model Machine Learning XGBoost Classifier melalui Teknik Oversampling dalam Prediksi Penyakit AIDS, JURNAL MEDIA INFORMATIKA BUDIDARMA. Diakses dari https://ejurnal.stmik-budidarma.ac.id/mib/article/view/7501

[14] KLASIFIKASI DAN EVALUASI PERFORMA MODEL RANDOM FOREST UNTUK PREDIKSI STROKE, JT : Jurnal Teknik, Okt. 2023. Diakses dari https://jurnal.umt.ac.id/index.php/jt/article/download/9099/4575

[15] Adhitya, Y., et al. (2021). House Price Prediction Using Machine Learning with Streamlit Framework. Journal of Big Data.

[16] Garcia, L., & Martinez, R. (2022). Comparative Analysis of Streamlit, Dash, and Flask for Rapid Data App Development. IEEE Access.

[17] Chen, X., et al. (2020). Performance Evaluation of Scikit-learn Algorithms in Medical Diagnosis. Journal of Healthcare Informatics.

[18] Smith, J., & Brown, A. (2021). Enhancing Weather Prediction with Scikit-learn

Preprocessing Techniques. Environmental Data Science.

[19] Lee, S., & Park, H. (2020). Visualizing Text Classification Results for Improved Model Interpretation. Data Visualization Journal.

[20] Wang, Y., et al. (2022). Explainable AI with SHAP and Matplotlib for Deep Learning Models. AI & Society.

[21] Kumar, R., & Sharma, P. (2021). Plant Disease Detection Using ML and Streamlit. Precision Agriculture Journal.

[22] Zhang, L., et al. (2023). Interactive Visualization in Stock Prediction Web Apps. Financial Analytics Journal.

[23] Shneiderman, B. (2020). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. IEEE Transactions on Visualization and Computer Graphics.

[24] Lundberg, S., & Lee, S. (2017). A Unified Approach to Interpreting Model Predictions. NeurIPS.