Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.2025128472
Vol. 12, No. 2, April 2025, hlm. 251-262 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 36/E/KPT/2019 e-ISSN: 2528-6579

PENERAPAN FEATURE ENGINEERING DAN HYPERPARAMETER TUNING UNTUK MENINGKATKAN AKURASI MODEL RANDOM FOREST PADA KLASIFIKASI RISIKO KREDIT

Nadea Putri Nur Fauzi[*1], Siti Khomsah[2], Aditya Dwi Putra Wicaksono[3]

[1,2,3] Program Studi Sains Data, Telkom University, Purwokerto
Email: [1]20110031@ittelkom-pwt.ac.id, [2]sitijk@telkomuniversity.ac.id, [3]adityaw@telkomuniversity.ac.id
[*]Penulis Korespondensi

(Naskah masuk: 09 Januari 2024, diterima untuk diterbitkan: 10 April 2025)

Abstrak

Risiko kredit adalah hal yang penting untuk dianalisis di awal pengajuan kredit guna mengurangi nilai *Non-Performing Loan* (NPL) atau risiko gagal bayar. Pola pengetahuan risiko kredit bisa diketahui dari data-data historikal sehingga data pengajuan kredit baru bisa ketahui risikonya lebih awal. Pada penelitian-penelitian terdahulu, model klasifikasi untuk risiko kredit menggunakan *Random Forest* banyak ditemukan namun tidak mendalam dalam penerapan *preprocessing* dan akurasinya masih rendah. Maka penelitian ini bertujuan meningkatkan akurasi model klasifikasi algoritma *Random Forest* dengan menerapkan *tuning* parameter dan *feature engineering* yang lebih dalam. Metodologi penelitian yang digunakan adalah *Sample, Explore, Modify, Models*, dan *Assess* (SEMMA). Penelitian ini menerapkan berbagai kombinasi parameter dan menerapkan *feature engineering* untuk memperbaiki kualitas data. *Feature engineering* yang digunakan meliputi *oversampling* dan standardisasi. *Hyperparameter tuning* model Random Forest menggunakan metode *Random Search* dan *Grid Search* untuk mencari parameter paling optimal. Dataset penelitian adalah data sekunder (*Credit Risk*) yang terdiri dari 32.581 baris, 11 variabel prediktor dan 1 variabel respon. Hasil penelitian menunjukkan penerapan *feature engineering* signifikan meningkatkan akurasi model *Random Forest*, meningkat dari 92,56% menjadi 97,94% setelah menerapkan *oversampling* dan *standarisasi*. Sedangkan *hyperparameter tuning* tidak begitu signifikan meningkatkan akurasi model yang dibangun menggunakan dataset yang sudah dikenakan *preprocessing* maupun *feature engineering* dengan baik.

Kata kunci: *risiko kredit, klasifikasi, Random Forest, hyperparameter tuning, feature engineering*

APPLICATION OF FEATURE ENGINEERING AND HYPERPARAMETER TUNING TO IMPROVE THE ACCURACY OF RANDOM FOREST MODELS ON CREDIT RISK CLASSIFICATION

Abstract

Credit risk analysis is essential for minimizing the value of non-performing loans (NPL). Using historical data to understand credit risk patterns can help identify risks early in new credit applications. Previous research has often used Random Forest classification models for credit risk but found the need for more comprehensive preprocessing of applications and higher accuracy. This research aims to improve the accuracy of the Random Forest algorithm classification model by implementing parameter tuning and feature engineering. The SEMMA (Sample, Explore, Modify, Model, and Assess) methodology is used, which explores different parameters and feature engineering combinations to enhance data quality. Feature engineering techniques, such as oversampling and standardization, are applied. Hyperparameter tuning of the Random Forest model involves using Random Search and Grid Search methods to identify the optimal parameters. The research dataset, consisting of 32.581 lines, 11 predictor variables, and one response variable, is secondary data on Credit Risk. Results show that the application of feature engineering significantly improves the accuracy of the Random Forest model, increasing from 92,56% to 97,94% after applying oversampling and standardization. However, hyperparameter tuning does not significantly increase the accuracy of models built using well-preprocessed datasets or feature engineering.

Keywords: *credit risk, classification, Random Forest, hyperparameter tuning, feature engineering,*

1. PENDAHULUAN

Risiko kredit merujuk pada kemungkinan peminjam gagal melunasi pinjaman atau bunga terkait, yang mengakibatkan klasifikasi pinjaman sebagai *Non-Performing Loan* (NPL). NPL, yang didefinisikan sebagai pembayaran tertunda lebih dari 90 hari dapat berdampak signifikan pada aspek operasional dan pendapatan bank atau lembaga keuangan. Salah satu keberhasilan operasional bisnis sebuah bank atau lembaga keuangan dapat dilihat dari kemampuannya dalam memberikan kredit dengan meminimalisir risiko yang dihadapi (Setiawan and Pratama, 2019).

Salah satu pendekatan efektif untuk mengurangi risiko kredit melibatkan pemanfaatan data historis peminjaman melalui teknik *data mining* (Religia, Nugroho and Hadikristatnto, 2021). Data mining mencakup kategori prediktif dan deskriptif. Pada metode prediktif dapat dilakukan dengan pembuatan model klasifikasi. Klasifikasi yaitu pengelompokan objek yang mempunyai kesamaan karakteristik atau ciri ke dalam suatu kelas. Model klasifikasi bisa dibangun dengan mempelajari pola data masa lalu, salah satunya dengan memanfaatkan algoritma *machine learning*.

Beberapa algoritma *machine learning* yang dapat digunakan untuk mengklasifikasikan risiko kredit diantaranya *Naïve Bayes*, *Support Vector Machine* (SVM), *K-Nearest Neighbors*, dan *Decision Tree*. *Naïve Bayes* memiliki kelebihan dapat diimplementasikan secara cepat dan tidak membutuhkan banyak data pelatihan atau *training*, namun algoritma tersebut memiliki kekurangan saat datanya tidak saling independen dan mengabaikan hubungan antar fitur dan seleksi fitur. (Muhamad et al., 2017). *Support Vector Machine* (SVM) memiliki kelebihan yang dapat diandalkan saat digunakan karena berfungsi dengan mengoptimalkan batas pemisah yang optimal (*hyperplane)* dan kuat akan data yang berdimensi tinggi (Nugroho, Witarto and Handoko, 2003). Namun, algoritma ini kurang tepat diterapkan saat data yang digunakan berskala besar karena membutuhkan waktu proses yang lama. Selanjutnya terdapat algoritma *K-Nearest Neighbors* yang mudah diimplementasikan dengan menghitung jarak antar kelasnya, namun algoritma ini memiliki kekurangan yaitu sensitif terhadap data pencilan (*outliers*)(Nasri and Aw, 2020). Selain itu terdapat algoritma *Decision Tree* yang menggunakan konsep pohon dengan kelebihan mampu menangani data campuran diskrit dan kontinyu, namun lemah pada data tidak seimbang karena lebih mengutamakan kelas mayoritas dalam pembentukan pohon (Pratiwi and Arifin, 2024).

Pemilihan algoritma *machine learning* harus disesuaikan dengan karakteristik data yang ada. Data kredit mengandung beberapa informasi seperti pendapatan, pekerjaan, jumlah pinjaman, suku bunga, kepemilikan rumah, dan beberapa informasi lainnya. Selain itu, data kredit juga memiliki berbagai jenis tipe data, termasuk kategorikal, diskrit, dan kontinyu.

Adanya keberagaman tipe data, ukuran data yang besar, dan terdapat banyak data *outliers*, maka algoritma *Random Forest* digunakan sebagai metode prediksi yang sesuai untuk kasus ini karena keunggulannya dalam menangani kumpulan data kompleks dengan berbagai jenis fitur. *Random Forest* merupakan algoritma *ensemble* yang menggunakan kumpulan pohon keputusan acak untuk memperoleh prediksi yang akurat dan meminimalkan *overfitting*. Selain kuat terhadap *overfitting* (Sanjaya et al., 2020), *Random Forest* memiliki kinerja yang lebih baik dibandingkan dengan algoritma lain seperti *Support Vector Machine* (SVM) dan *Discriminant Analysis* (Liaw and Wiener, 2002).

Algoritma *Random Forest* memiliki beberapa parameter yang mempengaruhi kinerjanya. Penggunaan pohon yang terlalu sedikit membuat model mengalami kelemahan, begitupun pohon yang terlalu banyak (Probst, Wright and Boulesteix, 2019). Pada penelitian sebelumnya penggunaan algoritma *Random Forest* pada klasifikasi terkait kredit belum menerapkan *hyperparameter tuning* sehingga kurang menjelajahi parameter terbaik yang dapat digunakan (Prasojo and Haryatmi, 2021). Tidak adanya analisis mengenai parameter terbaik pada model *Random Forest* menyebabkan kualitas model yang dihasilkan kurang optimal dan mengurangi kemampuannya dalam mengidentifikasi risiko kredit secara akurat. Penelitian (Sunarya and Haryanti, 2022) menjelaskan pentingnya penggunaan algoritma optimasi seperti *Grid Search* dan *Random Search* dalam meningkatkan akurasi model. Dipaparkan juga pada penelitian (Khomsah, Cahyana and Aribowo, 2023) bahwa penggunaan *Grid Search* dan *Random Search* pada model *Random Forest* kuat terhadap data yang memiliki banyak *noise* dan anomali.

Pada data yang tidak seimbang atau *imbalanced* perlu dilakukan *feature engineering* untuk memperbaiki kualitas data. Adanya kelas data yang tidak seimbang dapat meningkatkan risiko *overfitting* dan representasi terbatas pada kelas minoritasnya. Penelitian (Wang et al., 2017) menunjukkan bahwa penerapan *feature engineering* khususnya *feature selection* dapat meningkatkan akurasi model *Machine learning* pada klasifikasi risiko kredit. Oleh karena itu, penelitian ini bertujuan untuk mengatasi kekurangan tersebut dengan menerapkan *feature engineering* dan *hyperparameter tuning* yang lebih cermat guna menciptakan model klasifikasi yang lebih baik.

Beberapa penelitian *Random Forest* pada domain lain (Xia et al., 2015; Speiser et al., 2019) menunjukkan bahwa penerapan seleksi fitur dan reduksi ruang fitur juga dapat meningkatkan akurasi model *Random Forest*. Teknik-teknik tersebut berhasil mengoptimalkan performa model dengan memilih fitur-fitur yang paling informatif dan mengatur parameter-parameter yang tepat, sehingga

meningkatkan prediksi dan keandalan model secara signifikan.

Penelitian (Prasojo and Haryatmi, 2021) menerapkan model *Random Forest* pada kasus klasifikasi *german credit* dengan kondisi data yang terdapat banyak fitur dan kelas target tidak seimbang (*imbalanced*). Tujuan dari penelitian ini adalah untuk mengetahui penerapan dan hasil akurasi terbaik menggunakan algoritma *Random Forest*. Hasil penelitian ini menunjukkan bahwa akurasi model *Random Forest* yang didapat tergolong baik dengan akurasi 83%.

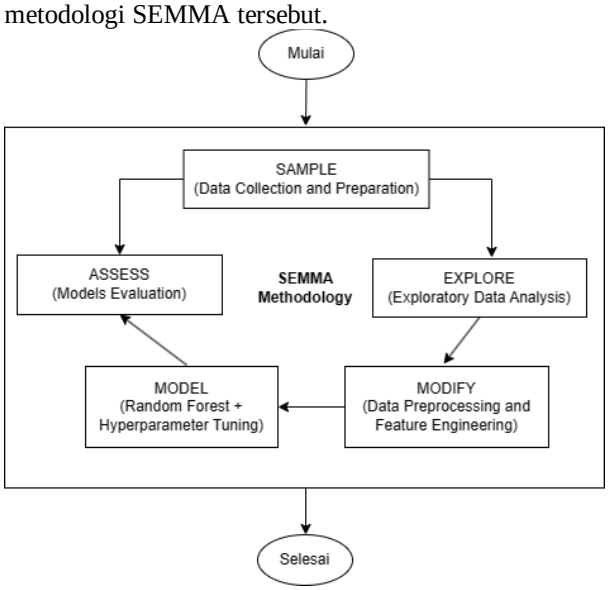
Penelitian selanjutnya (Sanjaya et al., 2020) membandingkan akurasi model *Random Forest*, DNN, dan *Adaboost* dengan menerapkan *hyperparameter tuning* pada data yang memiliki banyak fitur dan *imbalanced*. Hasil dari penelitian ini adalah model *Random Forest* mendapatkan akurasi terbaik sebesar 90,63% dibandingkan kedua model lainnya. Sedangkan penelitian tentang penerapan *Random Forest* dengan *hyperparameter tuning* pada klasifikasi risiko kredit pada data yang besar, berdimensi tinggi, dan *imbalanced* berhasil mendapatkan akurasi yang sangat baik (99,67%) dan nilai AUC yang sangat baik (0,99) (Uddin et al., 2022).

Penelitian selanjutnya (Sunarya and Haryanti, 2022) yang membandingkan kinerja 3 algoritma optimasi pada *Random Forest* yaitu *Grid Search*, *Random search*, dan *Bayesian Search* pada dataset *heart failure clinical records data*. Hasil penelitian tersebut menunjukkan bahwa metode *Random Search* mendapatkan akurasi tertinggi dibandingkan 2 metode lainnya yaitu *Grid Search* dan *Bayesian Search.* Akurasi model yang didapat yaitu 85,63%. sedangkan *Grid Search* untuk pengoptimalan parameter model *Random Forest* pada data *imbalanced* berhasil mendapatkan akurasi yang sangat baik yaitu 97% (Zhao, Hou and Ran, 2022). Metode penanganan *imbalanced data* seperti *random oversampling* juga pernah diteliti, pada data risiko kesehatan ibu hamil dari UCI *machine learning*, hasilnya akurasi *Random Forest* meningkat (Aryanti, Misriati and Hidayat, 2023).

Berdasarkan beberapa penelitian sebelumnya tersebut, maka penelitian ini menerapkan *feature engineering* untuk meningkatkan kualitas data melalui pemilihan fitur relevan, standardisasi, dan penanggulangan data yang tidak seimbang. Selain itu, *hyperparameter tuning* akan dilakukan untuk mengoptimalkan kinerja algoritma *Random Forest* dengan menganalisis berbagai kombinasi parameter. Dengan mengintegrasikan *feature engineering* dan *hyperparameter tuning* secara komprehensif, penelitian ini bertujuan menghasilkan model klasifikasi risiko kredit yang lebih akurat, memberikan manfaat bagi lembaga keuangan dalam membuat keputusan kredit yang lebih baik dan mengurangi dampak risiko kredit yang tinggi.

2. METODE PENELITIAN

Alur penelitian yang dilakukan menggunakan metodologi SEMMA. Metodologi SEMMA adalah salah satu metode data mining yang sering digunakan di berbagai penelitian. SEMMA merupakan akronim dari *Sample*, *Explore*, *Modify*, *Model*, dan *Assess*(Kurniawan et al., 2022). Diagram pada Gambar 1 menunjukkan alur penelitian menggunakan metodologi SEMMA tersebut.



Gambar 1. Alur Penelitian

2.1. *Sample*

Sample merupakan tahapan pemilihan data yang relevan. Data yang digunakan pada penelitian ini berupa data risiko kredit, merupakan dataset sekunder berasal dari kaggle berjudul "Credit Risk" oleh Lao Tse(Tse, 2020) yang tediri dari 32.581 baris dan 12 fitur. Tabel 1 merupakan penjelasan mengenai fitur data.

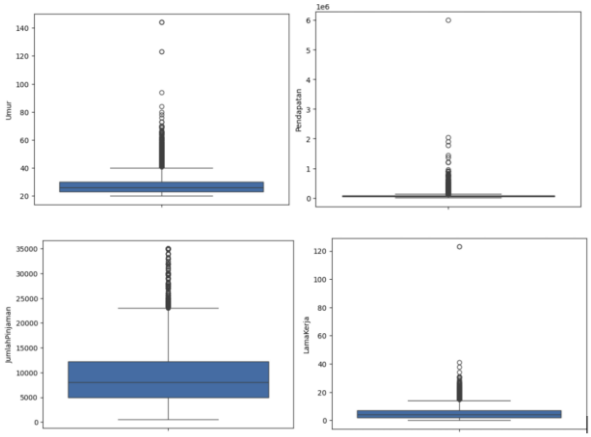
Tabel 1. Fitur Data

Fitur Data	Tipe Data
Umur	Integer
Pendapatan	Integer
KepemilikanRumah	Kategorikal
LamaKerja	Integer
TujuanPeminjaman	Kategorikal
HasilPemeriksaanBackground	Kategorikal
JumlahPinjaman	Kategorikal
SukuBunga	Integer
%Pendapatan	Integer
HistoryKegagalan	Kategorikal
JumlahHistoriPeminjaman	Integer
StatusPinjaman	Kategorikal(0= tidak berisiko, 1= berisiko)

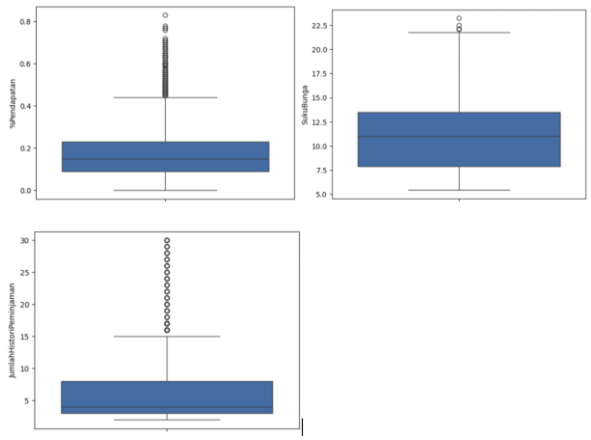
2.2. *Explore*

Explore merupakan tahapan kedua pada metodologi SEMMA yaitu mengeksplorasi data secara mendalam. Pada tahapan ini dilakukan *Exploratory Data Analysis* (EDA) meliputi ukuran data, informasi fitur, statistik deskriptif, korelasi fitur, *missing value*, dan anomali. Berdasarkan hasil EDA, dataset *Credit Risk* berisi 12 variabel terdiri 7 variabel

bertipe numerik dan 4 bertipe *string*, dan 1 variabel kelas. Analisis distribusi data menggunakan *Box and Whisker Plots* atau visualisasi *Box Plot* untuk menunjukkan data terdistribusi tidak normal pada semua variabel numerik, ditunjukkan Gambar 2-Gambar 3.



Gambar 2. BoxPlot Variabel Umur, Pendapatan, Jumlah Pinjaman, Lama Kerja



Gambar 3. BoxPlot Variabel Persentase Pendapatan, Suku Bunga, Jumlah Pinjaman

Langkah berikutnya deteksi *missing value* menggunakan perintah *isnull* pada *library* pandas, hasilnya ditemukan 4.011 data yang kosong (*null*). Variabel yang kosong yaitu pada LamaKerja sebanyak 895 baris dan 3.116 baris pada variabel SukuBunga, ditunjukkan Tabel 2. Persentase data *null* terlihat masih dibawah 1%, hal ini menjadi pedoman untuk penanganan *missing value* yang akan dibahas pada sub bagian *modify*.

Tabel 2. Variabel dengen Nilai Null/NaN

Nama Kolom	Jumlah Data	Persentase (%)
LamaKerja	895	0,027
SukuBunga	3.116	0,095
Total missing value	**4.011**	

Berikutnya terkait eksplorasi data *outlier*. Data dikatakan sebagai *outliers* jika berada di bawah *lower bound* dan di atas *upper bound* (Lestari, 2023). Kedua nilai tersebut dihitung dengan persamaan (1) dan persamaan (2) (Lestari, 2023). Hasilnya pada Tabel 3, jumlah *outlier* masih dibawah 1% untuk semua variabel numerik.

$$Upper\ Bound = \text{Q3} + (1.5 \text{ x IQR}) \quad (1)$$

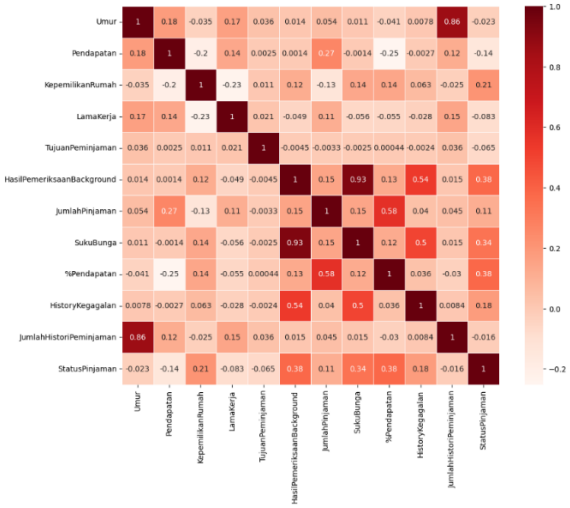
$$Lower\ Bound = \text{Q1} - (1.5 \text{ x IQR}) \quad (2)$$

Tabel 3. Data Outlier

Nama Kolom	Jumlah Outlier	Persentase (%)
Umur	1494	0,046%
Pendapatan	1484	0,045%
LamaKerja	853	0,025%
JumlahPinjaman	1689	0,051%
SukuBunga	6	0,0001%
%Pendapatan	651	0,019%

Data *outliers* yang tertera pada Tabel 3 tidak cukup banyak pada variabel Umur, Pendapatan, Lama Kerja, Jumlah Pinjaman, dan persentase pendapatan. Namun algoritma yang digunakan yaitu *Random Forest* tahan terhadap data *outliers*. Pada tahap ini juga dilakukan deteksi data tidak biasa atau anomali. Data yang dianggap tidak biasa adalah data pada kolom LamaKerja yang nilainya 123. Selain itu juga menerapkan filter pada kolom Umur bahwa hanya umur 21 s/d 65 tahun yang digunakan pada penelitian ini, dikarenakan umur peminjam jarang yang diatas usia 65 tahun dan dibawah 21 tahun. Hasilnya, ada 52 data yang termasuk data anomali.

Selanjutnya analisis korelasi antar fitur menggunakan korelasi metode *pearson correlation*. Visualisasi korelasi fitur dengan *heatmap* (Gambar 4) menunjukkan variabel yang memiliki korelasi tinggi yaitu: Umur dengan JumlahHistoriPeminjaman dan HasilPemeriksaanBackground dengan SukuBunga.

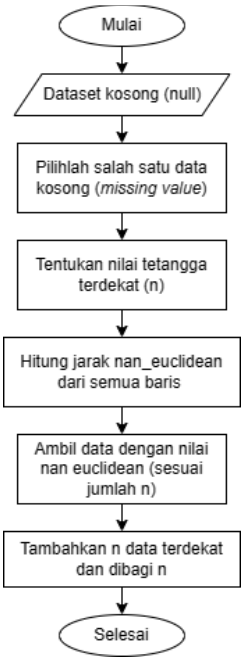


Gambar 4. Korelasi Antar Variabel

2.3. *Modify*

Modify merupakan tahapan ketiga pada metodologi SEMMA yang melibatkan teknik *data preprocessing* antara lain *handling missing value*, *replace value*, *encoding* dan *feature engineering* yang meliputi *oversampling* dan standardisasi.

Salah satu teknik untuk menangani *missing value* adalah KNNImputer. KNNImputer adalah metode imputasi data yang menggunakan algoritma *K-Nearest Neighbors* (KNN) untuk mengisi nilai-nilai yang hilang dalam sebuah dataset berdasarkan nilai-nilai tetangga terdekatnya (Widianti and Pratama, 2024). Alur kerja KNNImputer terdapat pada Gambar 5.

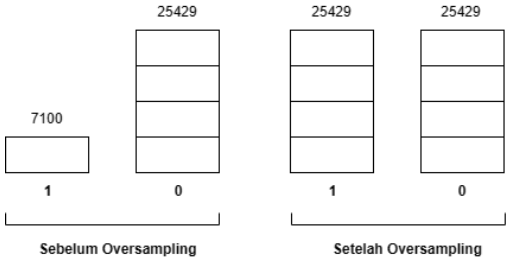


Gambar 5. Alur Kerja KNNImputer

Selanjutnya yaitu tahapan untuk mengatasi data anomali dan data tidak biasa. Data tidak biasa yang bernilai 123 akan dihapus. Data dikatakan anomali jika nilainya sangat jauh dari rata-rata data lainnya. Data yang terdeteksi anomali akan dihapus dari kumpulan data lainnya. Data anomali yang ditemukan yaitu pada variabel Umur dibawah 21 tahun dan diatas 65 tahun, yang jauh dari rata-rata, sehingga perlu hapus. Keberadaan anomali Umur diatas 65 tahun bisa jadi karena pada umur tersebut sudah pensiun (bagi pekerja) sedangkan umur dibawah 21 tahun jarang yang sudah bekerja. Penghapusan data anomali menyisakan data 32.529, kemudian digunakan pada proses selanjutnya.

Langkah selanjutnya yaitu mengganti nilai- nilai yang kurang tepat sebelum pemodelan (*Replace Value*). Nilai yang kurang sesuai pada data akan diganti menggunakan fungsi 'replace' dan melakukan perhitungan manual untuk mendapatkan nilai yang benar. Sedangkan penanganan *outlier* pada kasus ini, data *outlier* tidak dihapus karena sangat banyak, ini untuk menghindari bias pada data.

Setelah tahapan *preprocessing data* selesai, maka dilanjutkan dengan *feature engineering (FE)*. Tahapan pertama pada *feature engineering* yaitu *encoding* data. *Encoding* merupakan pengubahan data teks menjadi numerik sehingga data dapat dikenali oleh mesin. Penelitian ini menggunakan label *encoder* agar tidak menambah jumlah fitur pada data. Tahap selanjutnya yaitu *oversampling*. *Oversampling* merupakan teknik menyeimbangkan data dengan memperbanyak data pada kelas minoritas (Naldi and Agustian, 2023). Metode *oversampling* yang digunakan adalah *Random Over Sampling* (ROS) yang akan menduplikasi data minoritas menjadi seimbang dengan data mayoritas (Wongvorachan, He and Bulut, 2023). Gambar 6 merupakan ilustrasi dataset setelah dilakukan *oversampling*.



Gambar 6. Ilustrasi Data Hasil Oversampling

Setelah data dinyatakan seimbang (*balanced*), tahapan terakhir yaitu standardisasi. Standardisasi merupakan metode yang dilakukan untuk membuat beberapa variabel data memiliki rentang nilai yang sama, tidak ada yang terlalu besar maupun terlalu kecil (Zheng and Casari, 2018). Salah satu teknik standardisasi yaitu *Robust Scaler*, merupakan salah satu teknik optimasi untuk mentransformasikan suatu nilai pada data dengan menggunakan nilai median dan kuartil (Kusnaidi, Gulo and Aripin, 2022). Perhitungan *Robust Scaler* pada persamaan (3).

$$x' = \frac{x - median(x)}{Q3 - Q1} \quad (3)$$

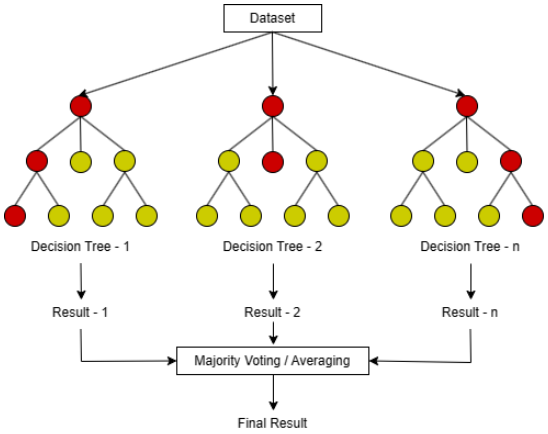
Dimana x' merupakan hasil standardisasi menggunakan *Robust Scaler*, Q3 – Q1 merupakan hasil *Interquartile Range* (IQR).

2.4. Model

Model merupakan tahapan keempat pada metodologi SEMMA yang melibatkan pembuatan model menggunakan algoritma *machine learning*. Pada tahapan ini akan dilakukan pemodelan menggunakan algoritma *Random Forest* dan dilakukan *hyperparameter tuning* untuk mencari parameter paling optimal. Algoritma *Random Forest* memiliki banyak parameter seperti kedalaman pohon, penentuan jumlah cabang, daun, dan jumlah pohon. Pemilihan nilai parameter yang tepat sebelum proses pelatihan akan membantu model belajar dari data.

Random Forest adalah salah satu algoritma *machine learning* yang bekerja dengan cara mengkombinasikan beberapa algoritma *Decision Tree* (George and Sumathi, 2020). Random Forest termasuk algoritma *ensemble* yang artinya algoritma ini membangun model di atas banyak model sehingga

mendorong model tersebut mendapatkan hasil lebih baik (Khomsah, 2021).



Gambar 7. Algoritma Random Forest

Gambar 7 merupakan contoh dari pengambilan keputusan algoritma *Random Forest*. Berikut merupakan penjelasan langkah- langkah cara kerja algoritma *Random Forest*:

1. Penentuan parameter *Random Forest* (jumlah pohon (*bootstrap*), kedalaman pohon maksimal, jumlah maksimal fitur, kriteria (gini atau entropi), jumlah minimal sampel agar node bisa terbagi, dan jumlah minimal sampel agar node menjadi daun).
2. Pembuatan *bootstrap- bootstrap* dari dataset.
3. Pembuatan pohon dari masing- masing *bootstrap*.
4. Prediksi kelas data baru berdasarkan pohon-pohon yang terbentuk.
5. Menentukan kelas data berdasarkan *majority voting*.

Sedangkan untuk mendapatkan parameter terbaik diterapkan *hyperparameter tuning* (HT) menggunakan teknik *Grid Search* dan *Random Search*. *Grid Search* merupakan metode pemilihan parameter terbaik dengan menjelajahi keseluruhan parameter yang diinputkan pada model. Berbeda dengan *Grid Search*, *Random Search* akan mengambil parameter-parameter yang diinputkan secara acak.

Terdapat 4 model yang dibuat dan dibandingkan dalam penelitian ini, seperti pada Tabel 4. Penjelasan lebih lanjut mengenai keempat model akan dijelaskan pada bagian hasil dan analisis.

Tabel 4. Skenario Pemodelan

Model	Keterangan
Model 1	Baseline Model
Model 2	FE + RF Default
Model 3	FE + HT Random Search
Model 4	FE + HT Grid Search

Keterangan Tabel 4:
FE : Feature Engineering
RF: Random Forest
HT: Hyperparameter tuning

2.5. Assess

Assess atau penilaian merupakan tahapan terakhir pada metodologi SEMMA yang melibatkan penilaian atau *evaluation*. Pada tahapan ini akan mengevaluasi hasil model *Random Forest* menggunakan *Accuracy*, *Confusion Matrix*, *Classification Report*, dan grafik *Area Under the Curve* (AUC). Akurasi dihitung dengan persamaan (4).

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

Dimana:

a. *True Positive* (TP) merupakan jumlah observasi yang benar- benar termasuk ke dalam kelas positif dan diprediksi benar oleh model.
b. *True Negative* (TN) merupakan jumlah observasi yang benar- benar termasuk ke dalam kelas negatif dan diprediksi dengan benar oleh model.
c. *False Positive* (FP) merupakan jumlah observasi yang diprediksi sebagai bagian dari kelas positif oleh model, tetapi sebenarnya observasi tersebut termasuk dalam kelas negatif.
d. *False Negative* (FN) merupakan jumlah observasi yang diprediksi sebagai bagian dari kelas negatif oleh model, tetapi sebenarnya observasi tersebut termasuk dalam kelas positif.

Untuk membuktikan nilai akurasi model yang baik, dapat dilihat dari nilai *Confusion Matrix* seperti Gambar 8.

		Predicted Class	
		Normal	Attack
Actual Class	Normal	True Negative (TN)	False Positive (FP)
	Attack	False Negative (FN)	True Positive (TP)

Gambar 8. *Confusion Matrix*

Model dikatakan baik jika nilai TN+TP > FN+FP. Ukuran kebaikan model juga akan dilihat dari nilai *precision*, *recall*, dan *f1-score*. Untuk melihat kinerja model dalam membedakan kelas biner (0 dan 1) maka akan dicari nilai AUC. Semakin tinggi nilai AUC maka semakin baik sebuah model yang dibuat (Widayati, Prihati and Widjaja, 2021).

3. HASIL DAN ANALISIS

Setelah tahap *Modify* yang dilakukan, dataset siap digunakan untuk pemodelan. Ukuran data awal yaitu kelas tidak berisiko (0) yaitu 7.100 sampel dan kelas berisiko (1) adalah 25.429 sampel. Setelah tahap modify ini nilai kosong sudah tidak ada, tidak ada data anomali, dan ukuran kedua kelas data

seimbang (*balanced*). Hasil *balancing* dengan *Random Oversampling* yaitu sebanyak 50.858.

Tahapan selanjutnya yaitu pemodelan. Terdapat 4 model yang akan dibandingkan. Namun sebelum ke tahap *modelling* terdapat tahapan pra pemodelan yaitu pembagian data independen dan dependen serta pembagian data (*training* dan *testing*). Data independen terdiri dari 11 variabel Umur, Pendapatan, KepemilikanRumah, LamaKerja, TujuanPeminjaman, HasilPemeriksaanBackground, JumlahPinjaman, SukuBunga, HistoryKegagalan, JumlahHistoriPeminjaman, dan Konfirmasi%Pendapatan. Sedangkan data dependen yaitu kolom target (StatusPinjaman).

Ukuran data *training* dan *testing* yang digunakan yaitu perbandingan 70:30, artinya 70% untuk data *training* dan 30% untuk data *testing*. Pembagian 70:30 dipilih karena rasio tersebut seringkali digunakan pada pemodelan *machine learning* pada dataset yang berukuran antara 100 sampai dengan 1.000.000 sampel (Muraina, 2022).

3.1. Model 1 (Baseline Model)

Model pertama menggunakan dataset awal (tanpa *feature engineering*) dan menggunakan model *Random Forest* dengan parameter *default*, pada Tabel 5.

Tabel 5. Parameter Model 1 (*Default Random Forest*)

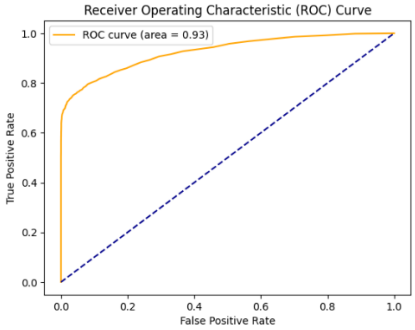
Parameter	Nilai
N_estimators	100
Criterion	Gini
Max_depth	None
Max_features	Sqrt
Min_samples_split	2
Min_samples_leaf	1

Pada model pertama ini hanya dilakukan penghapusan data kosong/*Null*/*NaN* sehingga data yang digunakan berjumlah 28.570 baris dengan 11 variabel prediktor dan 1 variabel respon. Dataset dibagi menjadi data *training* 70% (20.000) dan data *testing* 30% (8.571). Hasil *classification report* model 1 ditunjukkan Tabel 6.

Tabel 6. *Classification Report* Model 1

Kelas	Precision (%)	Recall (%)	F1-score (%)
0	92	99	95
1	95	69	80

Tabel 5 menunjukkan bahwa model *baseline* untuk kelas tidak berisiko (0) mempunyai presisi 92%, *recall* 99%, dan *f1_score* 95%. Sedangkan untuk kelas berisiko (1) memiliki presisi model 95%, namun nilai *recall* hanya 69%, dan *f1-score* 80%. Pada kelas berisiko (1) model tidak sensitif, ini ditunjukkan *recall* yang rendah. Namun jika dilihat nilai AUC sebesar 0,93 seperti pada Gambar 9, model dikatakan sudah baik.



Gambar 9. Grafik Kurva ROC dan Nilai AUC Model 1

3.2. Model 2 (FE + RF Default)

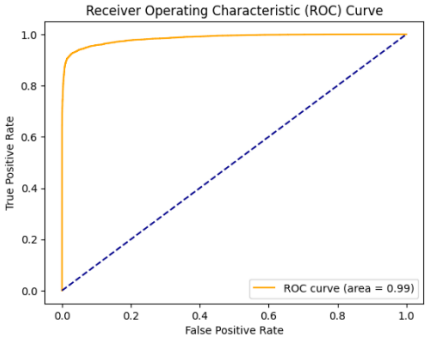
Model menerapkan 2 tahapan pada *feature engineering* yaitu *oversampling* dan standardisasi sehingga dataset yang digunakan menjadi seimbang antara kelas 0 dan kelas 1. Sedangkan parameter *Random Forest* menggunakan nilai *default*. Dataset yang digunakan adalah hasil *oversampling* berjumlah 50.858 baris dengan 12 kolom. Data awal sebelum *oversampling*, jumlah data kelas 0 sebanyak 25.429 dan kelas 1 sebanyak 7.100. Setelah penyeimbangan kelas 1 dengan *Random Oversampling* maka jumlah data kelas 1 sebanyak 25.429. Sehingga total data berjumlah 50.858.

Dataset dibagi 70% *training* (35.600) sedangkan data *testing* 30% (15.258). Model *Random Forest* yang digunakan yaitu model *default Random Forest*. *Classification report* model 2 ditunjukkan Tabel 7.

Tabel 7. *Classification Report* Model 2

Kelas	Precission (%)	Recall (%)	F1-score (%)
0	98	98	98
1	98	98	98

Model 2 memiliki performa presisi, *recall*, dan *f1-score* yang sama yaitu 98%. Ini menunjukkan bahwa setelah dilakukan oversampling, model lebih konsisten baik dari segi sensitivitas dan tingkat keakuratan yang baik. Dilihat dari nilai AUC sebesar 0,99 seperti pada Gambar 10, menunjukkan model sangat baik.



Gambar 10. Grafik Kurva ROC dan Nilai AUC Model 2

3.3. Model 3 (FE + HT Random Search)

Model ketiga dengan menerapkan *feature engineering* dan *hyperparameter tuning*

menggunakan *Randomized Search CV*. Dataset yang digunakan berjumlah 50.858 baris dengan 12 kolom. Data *training* berjumlah 70% sedangkan data *testing* 30%. *Feature engineering* yang diterapkan yaitu *oversampling* dan standardisasi. Parameter-parameter yang diuji coba (Tabel 8) antara lain jumlah pohon (N_estimator), fungsi kriteria (criterion), kedalaman pohon (Max_depth), maksimum fitur untuk membangun pohon (max_features), jumlah sampel minimum untuk membentuk node (min_samples_split), dan jumlah minimal sampel yang diperlukan untuk berada pada simpul daun (min_samples_leaf). Beberapa nilai kriteria tersebut diterapkan sebagai parameter *Random Forest.*

Tabel 8. Parameter Random Forest

Model	Keterangan
N_estimators	100, 200, 300, 400, 500, 600, 700, 800, 900, 1000
Criterion	Gini, Entropy, Logloss
Max_depth	None, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55
Max_features	None, sqrt, log2
Min_samples_split	2, 5, 10, 20
Min_samples_leaf	1, 2, 4, 8, 10

Tabel 9. Parameter Terbaik Random Search

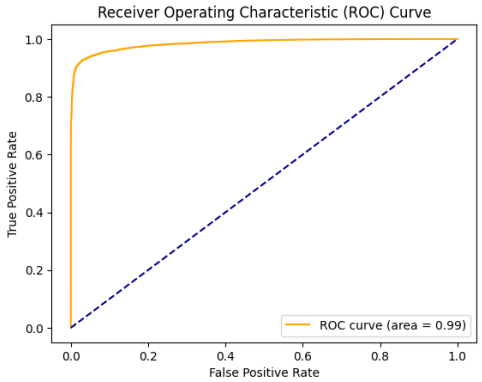
Model	Keterangan
N_estimators	900
Criterion	Gini
Max_depth	None
Max_features	sqrt
Min_samples_split	2
Min_samples_leaf	1

Dengan teknik *Random Search*, Tabel 9 menunjukkan hasil parameter terbaik untuk membangun model Random Forest yaitu jumlah pohon 900 dengan fungsi kriteria *Gini*, tanpa dibatasi jumlah kedalaman pohon, fitur maksimum menggunakan fungsi akar kuadrat (sqrt), jumlah minimal untuk membentuk node adalah 2 dan sampel minimal simpul daun adalah 1. Berdasarkan parameter terbaik yang didapat pada Tabel 9, *classification report* yang didapat ada pada Tabel 10.

Tabel 10. *Classification Report* Model 3

Model	Kelas	Precision (%)	Recall (%)	F1-score (%)
3	0	98	98	98
	1	98	98	98

Berdasarkan Tabel 10 menunjukkan bahwa *tuning* parameter tidak berpengaruh pada performa model. Model 3 ini memiliki tingkat presisi dan sensitvitas (*recall*) sama dengan model 2. Dengan demikian, pada kasus klasifikasi data risiko kredit dengan dataset Lao Tse ini, perlakuan *oversampling*, standardisasi, imputasi *missing value* sudah cukup menaikkan akurasi model. Performa model ini juga didukung oleh nilai AUC sebesar 0,99 yang sangat baik, pada Gambar 11.



Gambar 11. Grafik Kurva ROC dan Nilai AUC Model 3

3.4. Model 4 (FE + HT Grid Search)

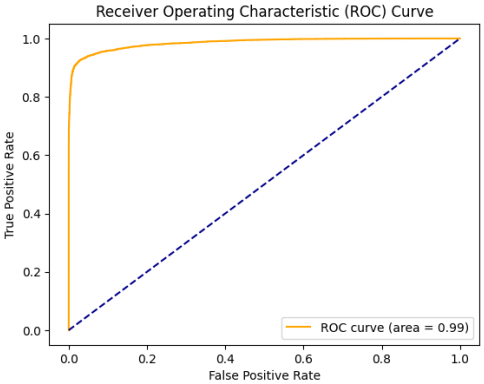
Model keempat dengan menerapkan *feature engineering* dan *hyperparameter tuning* menggunakan *Grid Search CV*. Dataset yang digunakan berjumlah 50.858 baris dengan 12 kolom. Data *training* berjumlah 70% sedangkan data *testing* berjumlah 30%. *Feature engineering* yang diterapkan yaitu *oversampling* dan standardisasi. Parameter yang di tuning sama dengan model 3 yang terdapat pada Tabel 3. *Best parameter* yang didapat menggunakan *Grid Search* terdapat pada Tabel 11.

Tabel 11. Parameter Terbaik Grid Search

Model	Keterangan
N_estimators	600
Criterion	Entropy
Max_depth	30
Max_features	sqrt
Min_samples_split	2
Min_samples_leaf	1

Berdasarkan parameter terbaik yang didapat pada Tabel 11, *classification report* model ditunjukkan Tabel 12. Terlihat bahwa model 4 ini menghasilkan tingkat akurasi maupun sensitifitasnya sama dengan model 3. Artinya, *tuning* parameter *Grid Search* ini tidak berpengaruh pada performa model. Kesimpulannya, *tuning* parameter menggunakan *Grid Search* maupun *Random Search* tidak memberikan efek yang berbeda pada akurasi model.

Nilai AUC sebesar 0,99 terlihat pada Gambar 12.



Gambar 12. Grafik Kurva ROC dan Nilai AUC Model 4

Tabel 12. *Classification Report* Model 3

Model	Kelas	Precision (%)	Recall (%)	F1-score (%)
3	0	98	98	98
	1	98	98	98

Secara keseluruhan dari 4 model yang dibuat, akurasi yang didapatkan terdapat pada Tabel 13.

Tabel 13. Akurasi Model

Model	Akurasi
Baseline Model	92,56%
FE + RF Default	97,81%
FE + HT (Random Search)	97,86%
FE + HT (Grid Search)	97,94%

Nilai akurasi tertinggi terdapat pada model 4 yang menerapkan *feature engineering* dan *hyperparameter tuning* menggunakan *Grid Search CV* (97,94%) tetapi *Grid Search* hanya sedikit lebih baik dalam meningkatkan akurasi model *Random Forest* dibandingkan dengan *Random Search*. Keduanya tidak berbeda signifikan. Dari Tabel 10 juga dapat dikatakan bahwa penggunaan *feature engineering* dapat meningkatkan akurasi model yang signifikan namun *tuning* parameter tidak begitu signifikan untuk peningkatan akurasi model ini. *Tuning* parameter menjadi tidak efektif, jika model sudah baik hanya dengan *preprocessing* yang tepat.

Sedangkan jika dilihat dari nilai rata-rata *precision, recall,* dan *f1-score* antara 4 model yang dibangun, model 2, 3, dan 4 memiliki nilai yang sama, ditunjukkan Tabel 14.

Tabel 14. Rata-Rata Presisi, Recall, F1-Score

Model	Precision	Recall	F1-Score
Baseline Model	95,31%	69,36%	80,29%
FE + RF Default	98%	98%	98%
FE + HT (Random Search)	98%	98%	98%
FE + HT (Grid Search)	98%	98%	98%

Dari tabel 14 terlihat bahwa nilai *precision, recall* dan *f1-score* pada *baseline* model berbeda sangat jauh, ini disebabkan oleh kondisi data *imbalanced*. Berbeda dengan ketiga model lainnya yang menerapkan *feature engineering*, nilai *precision, recall* dan *f1-score* yang didapat sama. Tabel 11 juga menunjukkan bahwa kondisi data yang seimbang berdampak pada nilai *precision, recall* dan *f1-score* pada data. Sedangkan untuk nilai *Area Under the Curve* (AUC) yang didapat, model 1 mendapatkan nilai 0,93. Sedangkan pada model 2, 3, dan 4 memiliki kesamaan nilai AUC yaitu 0,99. Dilihat dari nilai AUC, keempat model sebenarnya sudah sangat baik.

4. KESIMPULAN

Berdasarkan data yang digunakan (*Credit Risk Dataset*) dengan kondisi data yang memiliki tipe variabel yang berbeda-beda, ukuran yang besar, terdapat banyak *outliers*, banyak *missing value*, dan *imbalanced*, maka penggunaan *feature engineering* dapat meningkatkan akurasi model *Random Forest* secara signifikan. Model 1 (*Baseline Model*) yang tidak menerapkan teknik *feature engineering* dan *hyperparameter tuning* mendapatkan akurasi sebesar 92,56%. Kemudian model 2 yang hanya menerapkan *feature engineering* mendapatkan akurasi sebesar 97,81%. Selanjutnya model 3 dengan menerapkan *feature engineering* dan *hyperparameter tuning* menggunakan *Randomized Search CV* mendapatkan akurasi 97,86% dan model terakhir dengan menerapkan *feature engineering* dan *hyperparameter tuning* menggunakan *Grid Search CV* mendapatkan akurasi sebesar 97,94%. Dari hasil tersebut dapat disimpulkan bahwa penerapan *feature engineering* dapat meningkatkan akurasi model *Random Forest* secara signifikan pada kasus klasifikasi risiko kredit. Nilai AUC model *baseline* (model 1) yang awalnya sebesar 0,93 meningkat menjadi 0,99 (pada model 2, 3, 4). Berdasarkan nilai AUC yang semakin besar maka model yang dibangun dapat membedakan antara kelas 0 dan 1 dengan baik. Artinya penggunaan *feature engineering* dapat meningkatkan kemampuan model dalam membedakan antara kelas 0 dan 1 secara tepat. Meskipun demikian, *tuning* parameter tidak terlalu berpengaruh pada model jika dataset sudah mendapatkan perlakuan *preprocessing* yang tepat.

DAFTAR PUSTAKA

ARYANTI, R., MISRIATI, T. AND HIDAYAT, R., 2023. KLIK: Kajian Ilmiah Informatika dan Komputer Klasifikasi Risiko Kesehatan Ibu Hamil Menggunakan Random Oversampling Untuk Mengatasi Ketidakseimbangan Data. Media Online, [online] 3(5), pp.409–416. Available at: <https://djournals.com/klik>.

GEORGE, S. AND SUMATHI, B., 2020. Grid Search Tuning of Hyperparameters in Random Forest Classifier for Customer Feedback Sentiment Prediction. [online] IJACSA) International Journal of Advanced Computer Science and Applications, Available at: <www.ijacsa.thesai.org>.

KHOMSAH, S., 2021. Sentiment Analysis On YouTube Comments Using Word2Vec and Random Forest Sentimen Analisis pada Opini YouTube Menggunakan Word2Vec dan Random Forest. Jurnal Informatika dan Teknologi Informasi, 18(1), pp.61–72. https://doi.org/10.31515/telematika.v18i1.4493.

KHOMSAH, S., CAHYANA, N.H. AND ARIBOWO, A.S., 2023. Hyperparameter Tuning of Semi-Supervised Learning for Indonesian Text Annotation. International Journal of Advanced Computer Science and Applications, 14(9), pp.250–256.

https://doi.org/10.14569/IJACSA.2023.0140927.

KURNIAWAN, A., RIFA'I, A., NAFIS, M.A., SEFRIDA, N. AND PATRIA, H., 2022. Pemilihan Metode Predictive Analytics dengan Machine Learning untuk Analisis dan Strategi Peningkatan Kualitas Kredit Perbankan. Indonesian Journal of Applied Statistics, 5(1), p.1. https://doi.org/10.13057/ijas.v5i1.55483.

KUSNAIDI, M.R., GULO, T. AND ARIPIN, S., 2022. Penerapan Normalisasi Data Dalam Mengelompokkan Data Mahasiswa Dengan Menggunakan Metode K-Means Untuk Menentukan Prioritas Bantuan Uang Kuliah Tunggal. Journal of Computer System and Informatics (JoSYC), 3(4), pp.330–338. https://doi.org/10.47065/josyc.v3i4.2112.

LESTARI, M.E., 2023. Penerapan PCA (Principal Component Analysis) pada Deteksi Outlier untuk Data Text.

LIAW, A. AND WIENER, M., 2002. Classification and Regression by randomForest. [online] Available at: <http://www.stat.berkeley.edu/>.

MUHAMAD, H., PRASOJO, C.A., SUGIANTO, N.A., SURTININGSIH, L. AND CHOLISSODIN, I., 2017. Optimasi Naïve Bayes Classifier dengan Menggunakan Particle Swarm Optimization pada Data Iris. Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), 4(3), pp.180–184.

MURAINA, I.O., 2022. Ideal dataset splitting ratios in machine learning algorithms: general concerns for data scientists and data analysts. In: 7th International Mardin Artuklu Scientific Researches Conference. [online] Mardin, Turkey. pp.496–504. Available at: <https://www.researchgate.net/publication/358284895>.

NALDI, A. AND AGUSTIAN, S., 2023. Klasifikasi sentimen Vaksin Covid-19 menggunakan K-Nearest Neighbor Berdasarkan Word Embeddings Fasttext Pada Twitter. ZONAsi: Jurnal Sistem Informasi, 5(2), pp.323–333.

NASRI, E. AND AW, A.S., 2020. Aplikasi Seleksi Penentuan Nasabah Untuk Penjualan Barang Secara Kredit Dengan Algoritma K-Nearest Neighbor. SAINTEK:Jurnal Sains & Teknologi, 4(1), pp.1–11.

NUGROHO, A.S., WITARTO, A.B. AND HANDOKO, D., 2003. Support Vector Machine-Teori dan Aplikasinya dalam Bioinformatika 1. [online] Available at: <http://asnugroho.net>.

PRASOJO, B. AND HARYATMI, E., 2021. Analisa Prediksi Kelayakan Pemberian Kredit Pinjaman dengan Metode Random Forest. Jurnal Nasional Teknologi dan Sistem Informasi, 7(2), pp.79–89. https://doi.org/10.25077/teknosi.v7i2.2021.79-89.

PRATIWI, T.W. AND ARIFIN, T., 2024. Optimasi Decision Tree Menggunakan Particle Swarm Optimization untuk Klasifikasi Kesuburan pada Pria. SISTEMASI: Jurnal Sistem Informasi, [online] 10(1), pp.1–12. https://doi.org/10.32520/stmsi.v10i1.967.

PROBST, P., WRIGHT, M.N. AND BOULESTEIX, A.L., 2019. Hyperparameters and tuning strategies for random forest. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, https://doi.org/10.1002/widm.1301.

RELIGIA, Y., NUGROHO, A. AND HADIKRISTATNTO, W., 2021. Klasifikasi Analisis Perbandingan Algoritma Optimasi pada Random Forest untuk Klasifikasi Data Bank Marketing. Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 5(1), pp.187–192. https://doi.org/10.29207/resti.v5i1.2813.

SANJAYA, J., RENATA, E., BUDIMAN, V.E., ANDERSON, F. AND AYUB, M., 2020. Prediksi Kelalaian Pinjaman Bank Menggunakan Random Forest dan Adaptive Boosting. Jurnal Teknik Informatika dan Sistem Informasi, 6(1). https://doi.org/10.28932/jutisi.v6i1.2313.

SETIAWAN, R. AND PRATAMA, A.A.P., 2019. Modal, Tingkat Likuiditas Bank, NPL dan Pertumbuhan Kredit Perbankan Indonesia (Capital, Level of Liquidity, NPL and Lending Growth of Indonesian Banks). Matrik : Jurnal Manajemen, Strategi Bisnis dan Kewirausahaan, 13(1), pp.96–107. https://doi.org/10.24843/MATRIK:JMBK.2019.v13.i01.p10.

SPEISER, J.L., MILLER, M.E., TOOZE, J. AND IP, E., 2019. A comparison of random forest variable selection methods for classification prediction modeling. Expert Systems with Applications, 134, pp.93–101. https://doi.org/10.1016/j.eswa.2019.05.028.

SUNARYA, U. AND HARYANTI, T., 2022. Perbandingan Kinerja Algoritma Optimasi pada Metode Random Forest untuk Deteksi Kegagalan Jantung. Jurnal Rekayasa Elektrika, 18(4). https://doi.org/10.17529/jre.v18i4.26981.

TSE, L., 2020. Credit Risk Dataset. https://www.kaggle.com/datasets/laotse/credit-risk-dataset.

UDDIN, M.S., CHI, G., JANABI, M.A.M. AL AND HABIB, T., 2022. Leveraging random forest in micro-enterprises credit risk modelling for accuracy and interpretability. International Journal of Finance and Economics, 27(3),

pp.3713–3729. https://doi.org/10.1002/ijfe.2346.

WANG, S., FU, B., LIU, H., JIANG, Z., WU, Z. AND HSU, D.F., 2017. Feature Engineering for Credit Risk Evaluation in Online P2P Lending. International Journal of Software Science and Computational Intelligence, 9(2), pp.1–13. https://doi.org/10.4018/ijssci.2017040101.

WIDAYATI, Y.T., PRIHATI, Y. AND WIDJAJA, S., 2021. Analisis Dan Komparasi Algoritma Na Ve Bayes Dan C4. 5 Untuk Klasifikasi Loyalitas Pelanggan Mnc Play Kota Semarang. TRANSFORMTIKA, 18(2), pp.161–172.

WIDIANTI, A. AND PRATAMA, I., 2024. Penanganan Missing Values dan Prediksi Data Timbunan Sampah Berbasis Machine Learning. Rabit : Jurnal Teknologi dan Sistem Informasi Univrab, 9(2), pp.242–251. https://doi.org/10.36341/rabit.v9i2.4789.

WONGVORACHAN, T., HE, S. AND BULUT, O., 2023. A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining. Information (Switzerland), 14(1). https://doi.org/10.3390/info14010054.

XIA, J., LIAO, W., CHANUSSOT, J., DU, P., SONG, G. AND PHILIPS, W., 2015. Improving Random Forest With Ensemble of Features and Semisupervised Feature Extraction. IEEE Geoscience and Remote Sensing Letters, 12(7), pp.1471–1475. https://doi.org/10.1109/LGRS.2015.2409112.

ZHAO, W., HOU, J. AND RAN, Q., 2022. Analysis of Corporate Credit Risk Based on Random Forest and TOPSIS Models. Financial Engineering and Risk Management, 5(4), pp.30–37. https://doi.org/10.23977/ferm.2022.050405.

ZHENG, A. AND CASARI, A., 2018. Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists. O'Reilly Media, Inc.

Halaman ini sengaja dikosongkan