JNATIA Volume 3, Nomor 3, Mei 2025 p-ISSN: 2986-3929
Jurnal Nasional Teknologi Informasi dan Aplikasinya e-ISSN: 3032-1948

Penggunaan Algoritma C4.5 dan Random Forest guna Meningkatkan Efisiensi Klasifikasi Penyakit Stroke

I Agus Indra Dipta Prayoga[a1], I Gusti Agung Gede Arya Kadyanan[a2]

Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
[1]agusindra270704@gmail.com
[2]gungde@unud.ac.id

Abstract

Stroke is a very serious problem throughout the world. According to a report from the World Heart Organization (WHO), in 2022, more than 12.2 million, or one in four people aged 25 years will experience a stroke, and more than 7.6 million new stroke sufferers every year throughout the world. An irregular lifestyle is the main cause of someone having a stroke. Therefore, we need a system that can be used as a stroke classification or detection tool based on a person's disease history. The stroke disease data used in this study was obtained through Kaggle with a total of 5110 data. Based on the results of research that has been carried out using two algorithm models, namely the C4.5 algorithm and Random Forest. A combination of these two algorithms has been obtained which produces a stroke classification system with fairly good accuracy, with an accuracy value of 92.4%.

Keywords: *Stroke, C4.5 Algorithm, Random forest, Classification, HyperParameter Tuning*

1. Pendahuluan

Stroke telah menjadi perhatian serius dalam dunia kesehatan, bukan hanya di Indonesia, tetapi juga di seluruh dunia. Hingga saat ini, stroke merupakan salah satu penyebab utama kematian dan kecacatan di dunia[1]. Pola hidup yang tidak sehat merupakan penyebab utama terjadinya *stroke*. Oleh karena itu, deteksi dan klasifikasi *stroke* yang tepat sangat penting karena dapat memiliki dampak serius jika tidak ditangani dengan cepat. Maka dari itu, diperlukan sistem pendeteksi *stroke* yang dapat memberikan penanganan dan klasifikasi yang cepat dan akurat.

Dalam pengembangan sistem klasifikasi penyakit yang telah ada, penelitian tersebut menggunakan berbagai teknik klasifikasi seperti *Support Vector Machines (SVM), Neural Networks, Random Forest, Naive Bayes, C4.5, dan Gradient Boosting Machines*[2]. Misalnya, dalam penelitian yang dilakukan oleh Putri dengan judul "*Data Mining Implementation for Diabetes Risk Prediction Using C4.5 Algorithm*," peneliti dapat memperoleh akurasi 90% dari model[3]. Dengan demikian, dalam studi ini, sistem akan dikembangkan menggunakan dua metode yaitu algoritma C4.5 dan Random Forest untuk memperoleh hasil akurasi yang lebih baik.

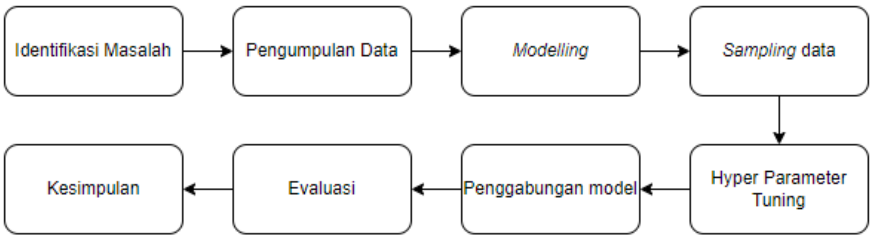
Pada penelitian ini akan digunakan *dataset* yang berisi *parameter input* seperti jenis kelamin, usia, riwayat penyakit pasien, tipe pekerjaan, tempat tinggal, nilai glukosa pasien, nilai bmi, dan status merokok. Model nantinya akan dievaluasi menggunakan matrik performa. Setelah itu, model akan digabungkan menjadi sebuah model baru yang diharapkan lebih efektif dari dua algoritma yang digunakan dalam mengklasifikasi penyakit *stroke*. Dengan dilakukannya penelitian ini, peneliti berharap dapat memperoleh wawasan mengenai akurasi pengukuran dari kedua metode sekaligus hasil penggabungan dari kedua metode yang digunakan untuk proses klasifikasi.

2. Metode Penelitian

Penelitian dibagi menjadi beberapa langkah yaitu yang pertama ada identifikasi masalah, proses pengumpulan data, proses modeling dari algoritma yang digunakan, *sampling data*, melakukan

p-ISSN: 2986-3929
e-ISSN: 3032-1948

proses *hyperparameter tuning*, evaluasi model, lalu model akan melalui proses penggabungan, dan langkah terakhir adalah pengambilan kesimpulan dari model yang sudah dibuat. Untuk gambaran dari tahapan penelitian dapat diperhatikan pada gambar 1 dibawah.



Gambar 1. Tahapan Penelitian

2.1. Pengumpulan Data

Data yang digunakan dalam penelitian ini berasal dari Kaggle (https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset). Tabel 1 di bawah ini memberikan penjelasan yang lebih rinci mengenai hal ini.

Tabel 1. Sejumlah 5 Data dari 5110 Dataset

	Id	Gender	Age	Hypertension	Hearth_disease	Glucose_level	BMI	Stroke
0	9064	Male	67	0	1	228.69	36.6	1
1	51676	Female	61	0	0	202.21	NaN	1
2	31112	Male	80	0	1	105.92	32.5	1
3	60182	Female	49	0	0	171.23	34.4	1
4	1665	Female	79	1	0	174.12	24.0	1

Data yang ditampilkan diatas berjumlah 5 baris dan 8 kolom data dari total 5110 baris data dan 12 kolom data, didalamnya terdapat beberapa input parameter seperti jenis kelamin, usia, riwayat penyakit pasien, tipe pekerjaan, tempat tinggal, nilai glukosa pasien, nilai bmi, dan status merokok.

a. Algoritma C4.5

Salah satu metode *data mining* yang dapat digunakan untuk mengklasifikasikan data *stroke* adalah klasifikasi menggunakan algoritma C4.5. Algoritma C4.5 dipilih dalam penelitian ini karena memiliki beberapa keunggulan yang sangat relevan untuk klasifikasi penyakit *stroke*. Sebagai evolusi dari algoritma ID3, C4.5 tidak hanya meningkatkan akurasi model melalui perhitungan nilai *entropy* dan *gain*, tetapi juga mampu menangani data yang lebih kompleks dengan lebih efisien[4]. Keuntungan utama dari menggunakan algoritma decision tree seperti C4.5 adalah kemampuannya untuk menyederhanakan proses pemecahan masalah yang kompleks, menjadikannya lebih mudah untuk diinterpretasikan oleh para peneliti dan praktisi medis. Ini sangat penting dalam konteks penyakit *stroke*, di mana memahami pola dan hubungan antara berbagai faktor risiko dapat membantu dalam pengambilan keputusan.

Algoritma C4.5 juga dapat menangani data dengan atribut kontinu dan atribut yang hilang, serta memiliki mekanisme untuk melakukan pemotongan pohon, yang membantu dalam mengurangi kompleksitas model dan mencegah pengklasifikasin berlebih. Dengan demikian, penggunaan algoritma C4.5 dalam penelitian ini diharapkan dapat menghasilkan model yang tidak hanya akurat tetapi juga mudah dipahami dan diimplementasikan, mendukung upaya dalam pencegahan dan penanganan penyakit *stroke* secara lebih

efektif [5]. Algoritma C4.5 memiliki dua rumus, yang pertama adalah rumus perhitungan nilai *entropy* dan rumus untuk mencari nilai *gain*.

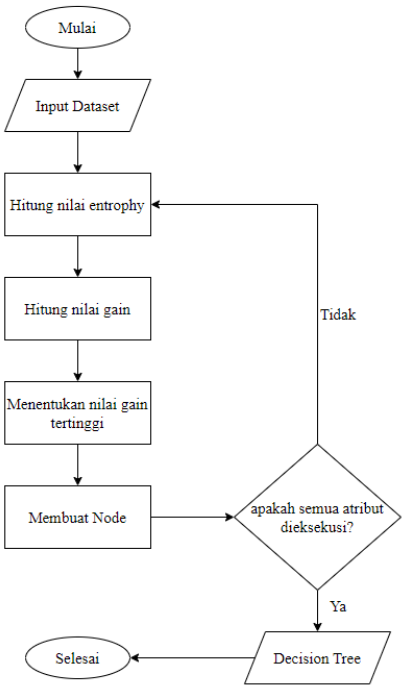
$$Entropy(i) = \sum_{j=1}^{m} f(i,j).log2f[(i,j)] \quad (1)$$

$i : Himpunan\ Kasus$
$m : Jumlah\ Partisi\ i$
$f(i,j) : Proporsi\ j\ Terhadap\ i$

$$gain\ (S.A) = Entropy(s) - \sum_{i}^{n} \frac{|si|}{|s|} * Entropy\ (si) \quad (2)$$

$S : Himpunan\ Kasus$
$A : Fitur$
$N : Jumlah\ Partisi\ Atribut\ A$
$|si| : Proporsi\ si\ terhadap\ S$
$|s| : Jumlah\ kasus\ Dalam\ S$

Adapun berikut merupakan gambaran dari proses dari algoritma C4.5, proses akan digambarkan melalui sebuah *flowchart* yang dapat dilihat pada gambar 2 dibawah ini.



Gambar 2. *Flowchart* Algoritma C4.5

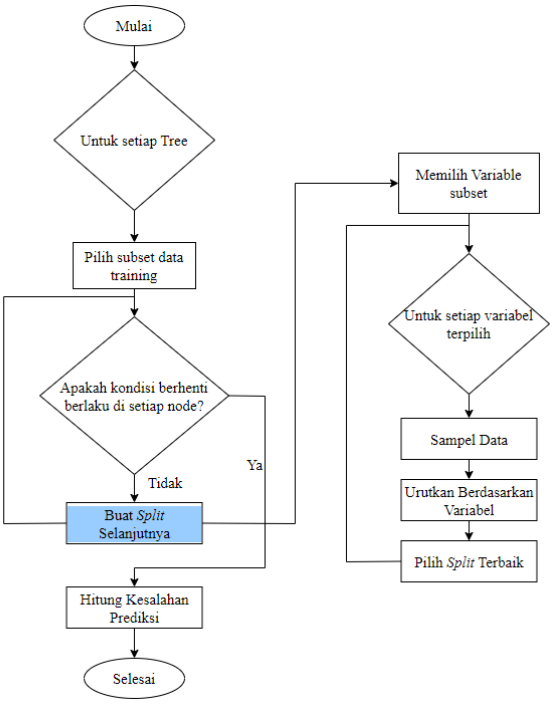
Flowchart diatas menggambarkan proses algoritma C4.5 untuk pembuatan *Decision Tree* dalam analisis data. Proses dimulai dengan menginput *dataset*, dilanjutkan dengan perhitungan nilai *entropy* untuk mengukur tingkat ketidakpastian dalam data. Kemudian, nilai *gain* dihitung untuk menentukan atribut yang paling baik memisahkan data. Atribut dengan nilai *gain* tertinggi dipilih untuk membuat *node*. Proses ini berulang, memeriksa apakah semua atribut telah dieksekusi. Jika belum, proses kembali ke perhitungan nilai *entropy* dan *gain*. Jika semua atribut sudah dieksekusi, *Decision Tree* selesai dibangun menggunakan algoritma C4.5 dan model siap untuk digunakan.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

b. Random Forest

Selanjutnya, diperlukan algoritma Random Forest dalam penelitian ini karena beberapa alasan yang krusial. Pemilihan algoritma Random Forest untuk klasifikasi penyakit *stroke* dalam penelitian ini didasarkan pada sejumlah keunggulan yang dimilikinya dalam menangani data medis yang kompleks dan bervariasi. Algoritma Random Forest mampu mengatasi masalah pengklasifikasian berlebih yang sering terjadi pada algoritma *decision tree* tunggal. Dengan membangun sejumlah besar pohon keputusan dan menggabungkan hasilnya, Random Forest meningkatkan akurasi prediksi sekaligus memberikan model yang lebih stabil. Teknik *sampling bootstrap* yang digunakan dalam algoritma ini memungkinkan model untuk mengeksplorasi berbagai *subset* data secara acak, sehingga mampu menangkap berbagai pola dan karakteristik data yang mungkin tidak terlihat jika hanya menggunakan satu *subset* data saja. Hal ini sangat penting dalam klasifikasi penyakit stroke, di mana faktor-faktor risiko dan gejala dapat sangat bervariasi antar individu[6].

Penggunaan *node-root, internal*, dan *leaf* dalam struktur *decision tree* memastikan bahwa setiap fitur dan variasi data dapat dievaluasi secara mendalam, memberikan model kemampuan untuk mengidentifikasi pola dan hubungan yang mungkin tidak terlihat melalui analisis sederhana. Dengan demikian, penerapan Random Forest dalam penelitian ini diharapkan dapat memberikan model prediktif yang efisien, yang tidak hanya bekerja baik pada data *training* tetapi juga pada data *testing*, sehingga dapat diaplikasikan secara efektif dalam berbagai situasi nyata. Pemodelan yang akurat ini penting untuk mendapatkan wawasan yang bermakna dari data dan untuk mendukung pengambilan keputusan yang berbasis data dalam berbagai aplikasi praktis[7]. Proses pembuatan struktur *decision tree* dari algoritma random forest dapat dilihat pada *flowchart* gambar 3 dibawah ini.



Gambar 3. *Flowchart* Pembuatan *Decision Tree* Random Forest

Flowchart diatas menggambarkan proses algoritma Random Forest untuk pembuatan model prediksi. Proses dimulai dengan pemilihan *subset data training* untuk setiap pohon (*Tree*). Kemudian, dipilih subset variabel untuk membentuk *decision tree*. Untuk setiap variabel terpilih, dilakukan sampling data, diurutkan berdasarkan variabel, dan dipilih *split* terbaik. Selanjutnya, akan dilakukan pemeriksaan apakah kondisi berhenti berlaku di

setiap node. Jika tidak, dibuat *split* selanjutnya. Proses ini berulang sampai kondisi berhenti terpenuhi. Setelah semua pohon terbentuk, kesalahan prediksi dihitung untuk mengevaluasi model. Proses berakhir ketika semua pohon selesai dibuat dan dievaluasi, menghasilkan sebuah model Random Forest yang utuh.

2.2. Modelling

Selanjutnya, Algoritma C4.5 dan Random Forest akan digunakan dalam proses *modeling*. Algoritma Random Forest adalah salah satu teknik *ensemble learning* yang menggunakan sejumlah *decision tree* untuk membuat prediksi, dan algoritma C4.5 adalah algoritma *decision tree* yang digunakan untuk membuat model prediktif berdasarkan data training[8]. Pada proses *modelling*, data yang digunakan akan dibagi menjadi 80% data *training* dan 20% data *testing.* Diharapkan pada penelitian ini menggunakan dua algoritma, dapat membuat model yang akurat dan dapat diandalkan untuk memprediksi berbagai kondisi atau kejadian berdasarkan data yang diberikan.

Modeling diperlukan dalam penelitian ini untuk beberapa alasan penting. *Modeling* memungkinkan untuk menangkap pola kompleks dan hubungan dalam data yang mungkin tidak terlihat secara langsung. Dengan menggunakan algoritma seperti C4.5 dan Random Forest, kita dapat membuat model prediktif yang mampu mengolah berbagai variabel input dan menghasilkan prediksi yang akurat. Selanjutnya, melalui proses pembelajaran dari *data training*, model ini dapat menggeneralisasi pola yang ditemukan sehingga mampu memberikan prediksi yang andal pada data baru yang belum pernah dilihat sebelumnya, ini sangat berguna dalam berbagai aplikasi praktis, mulai dari prediksi kejadian bisnis hingga diagnosis medis. Selain itu, penggunaan teknik seperti Random Forest dapat meningkatkan akurasi dan stabilitas prediksi dengan mengurangi risiko spesifikasi berlebihan yang mungkin terjadi jika hanya menggunakan satu *decision tree*. Oleh karena itu, pemodelan bukan hanya sekedar alat analisis, tetapi juga menjadi komponen krusial untuk memastikan bahwa hasil penelitian dapat diterapkan secara efektif dalam situasi nyata.

2.3. Sampling Data

Setelah melakukan *modeling* data, proses selanjutnya adalah melakukan *sampling* dari data yang sudah melakukan tahap *modeling*. Proses *sampling data* secara umum digunakan untuk mengubah pendistribusian dataset awal dengan cara memilih subset tertentu dari data yang sudah diberikan sebelumnya. Pada penelitian ini proses *sampling* akan menggunakan SMOTE, dalam penelitian ini SMOTE digunakan untuk mengatasi adanya ketidakseimbangan dalam data yang diberikan. SMOTE (*Synthetic Minority Over-sampling Technique*) merupakan sebuah metode yang digunakan dalam klasifikasi untuk menangani ketidakseimbangan kelas dengan membuat sampel sintetis dari kelas minoritas untuk menyamakan distribusi kelas, SMOTE beroprasi dengan cara membuat sebuah sampel sintetis data baru dari kelas minoritas yang dilakukan dengan cara menggabungkan titik-titik data yang sudah tersedia sebelumnya.

2.4. HyperParameter Tuning

Proses *hyperparameter tuning* dilaksanakan setelah melakukan *sampling* dari dataset yang diberikan. *Hyperparameter tuning* adalah sebuah proses pemilihan nilai optimal untuk parameter-parameter yang tidak dipelajari oleh model yang nantinya disebut sebagai *hyperparameter*. *Hyperparameter* ini ditentukan sebelum melatih model, peran *hyperparameter* cukup penting dalam menetapkan perilaku dan kompleksitas model[9]. Penyetelan *hyperparameter* berupaya menemukan kombinasi ideal yang memaksimalkan ukuran performa model, seperti *accuracy, precision, recall*, atau F1 *score*, dengan menelusuri rentang nilai *hyperparameter* secara sistematis. Langkah ini penting digunakan dalam penelitian ini untuk menyempurnakan model dan meningkatkan kapasitasnya untuk melakukan generalisasi terhadap masukan baru.

2.5. Penggabungan Model

Penggabungan dari kedua model dilakukan setelah proses *tuning* selesai. Penggabungan model bertujuan untuk memperoleh sebuah model baru yang memiliki karakteristik unggulan dari kedua model yang digunakan sebelumnya, yaitu algoritma C4.5 dan random forest. Karakteristik unggulan dari algoritma C4.5 adalah algoritma ini dapat membuat sebuah *decision tree* yang lebih optimal. Selain itu, algoritma C4.5 dapat digunakan untuk menangani data yang tidak seimbang. Selanjutnya, karakteristik unggulan dari algoritma random forest yaitu dapat mengurangi terjadinya spesifikasi yang berlebihan, dapat memilih atribut secara acak, kestabilan, dan skalabilitas yang tinggi. Dengan penggabungan kedua model pada penelitian ini diharapkan dapat menghasilkan model baru dengan karakteristik-karakteristik unggulan tersebut.

2.6. Evaluasi

Setelah semua proses pengolahan model selesai, hasil dari model yang sudah dibuat akan masuk ke tahapan evaluasi. Evaluasi akan dilakukan dengan mengamati hasil dari *metrics score* seperti *precision*, *recall,* F1, hasil dari *k-cross validation* dari masing-masing model. Adapun berikut merupakan beberapa persamaan *metrics* yang akan digunakan:

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$F1 = \frac{2 * P * R}{P + R} \tag{5}$$

$$TPR = \frac{TP}{TP + FN} \tag{6}$$

$$FPR = \frac{FP}{FP + TN} \tag{7}$$

$$akurasi = \frac{\sum data\ uji\ benar\ klasifikasi}{\sum total\ data\ uji} \times 100 \tag{8}$$

Keterangan:

TP: *True Positives* (merupakan prediksi model untuk seseorang mempunyai penyakit stroke dengan benar)
TN: *True Negatives* (merupakan prediksi model untuk seseorang tidak mempunyai penyakit stroke dengan benar)
FN: *False Negatives* (merupakan prediksi model untuk seseorang tidak mempunyai penyakit stroke namun seseorang mempunyai penyakit stroke)
FP: *False Positives* (merupakan prediksi model untuk seseorang mempunyai penyakit stroke namun seseorang tidak mempunyai penyakit stroke)
R : Nilai *Recall*
P : Nilai *Precision*

Menurut teori informasi, *precision* merupakan jumlah dokumen terkait yang didapatkan setelah kueri, sedangkan *recall* didefinisikan sebagai banyaknya dokumen relevan yang dikumpulkan setelah kueri selanjutnya dibagi dengan jumlah total dokumen relevan[10]. Penggunaan *K-fold cross validation* dalam proses evaluasi bertujuan untuk mendapatkan hasil penilaian akurasi yang lebih maksimal. Hal ini karena metode *k-fold cross validation* merangkum nilai rata-rata dari setiap iterasi yang tersedia. Selanjutnya, hasil TPR dan FPR dari setiap model akan direpresentasikan menggunakan kurva ROC. Kurva ROC (*Receiver Operating Characteristic*) merupakan sebuah grafik yang menampilkan hubungan antara sensitivitas (*True Positive Rate*) dan spesifisitas

(*False Positive Rate*) dari suatu model klasifikasi pada berbagai ambang batas pengambilan keputusan.

3. Hasil dan Pembahasan

Pada penelitian ini digunakan algoritma C4.5 dan Random Forest. Dataset yang digunakan dibagi menjadi 80% *training* dan 20% *testing*. Sebelum melakukan *data training*, dilakukan proses *sampling* menggunakan SMOTE terlebih dahulu. Selanjutnya dilakukan K-fold cross validation dengan pembagian data sebanyak 6 kali. Pembuatan model akan didasarkan pada data yang sudah di *sampling*. Kemudian, kedua model akan dilakukan *hyperparameter tuning* yang berguna meningkatkan performa keseluruhan dari model.

3.1. Hasil Evaluasi Model

a. Hasil Evaluasi Algoritma C4.5

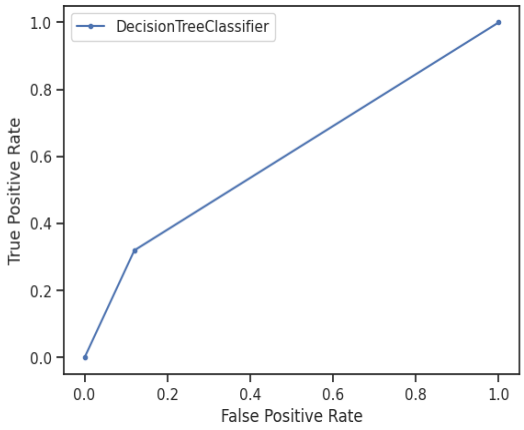
Berikut merupakan hasil evaluasi dari algoritma pembuatan *decision tree* C4.5. dapat dilihat pada gambar 2 hasil evaluasi *metrics score* dari algoritma ini menghasilkan akurasi sebesar 85%, akurasi *k-fold cross validation* menghasilkan sebesar 88,7%. Selanjutnya, dihasilkan nilai *Precision, Recall*, dan *F1 score* secara berturut-urut sebesar 92%, 85%, 88%. Hasil dari penilaian *metrics score* dari algoritma ini tergolong baik untuk proses klasifikasi. Untuk penjelasan lebih lengkap dapat dilihat pada gambar 2 dibawah ini.

```
DecisionTreeClassifier :
Accuracy Score: 0.8523

K-Fold Validation Mean Accuracy: 88.7376 %

Classification report:
              precision    recall  f1-score   support

           0       0.96      0.88      0.92       972
           1       0.12      0.32      0.17        50

    accuracy                           0.85      1022
   macro avg       0.54      0.60      0.55      1022
weighted avg       0.92      0.85      0.88      1022


Confusion matrix:
[[855 117]
 [ 34  16]]

Precision Score: 0.9206

Recall Score: 0.8523

ROC Score: 0.5998

F1 score: 0.8825
```

Gambar 4. *Metrics Score* Algoritma DecisionTree C4.5

Gambar diatas merupakan gambaran hasil penilaian *metrics score* dari algoritma C4.5. Selanjutnya akan diperlihatkan hasil dari perhitungan TPR dan FPR dari algoritma ini yang akan direpresentasikan dalam bentuk kurva ROC, berikut merupakan hasil dari perhitungan TPR dan FPR dalam kurva ROC algoritma C4.5.

p-ISSN: 2986-3929
e-ISSN: 3032-1948



Gambar 5. Kurva ROC Algoritma DecisionTree C4.5

b. Hasil Evaluasi Algoritma Random Forest

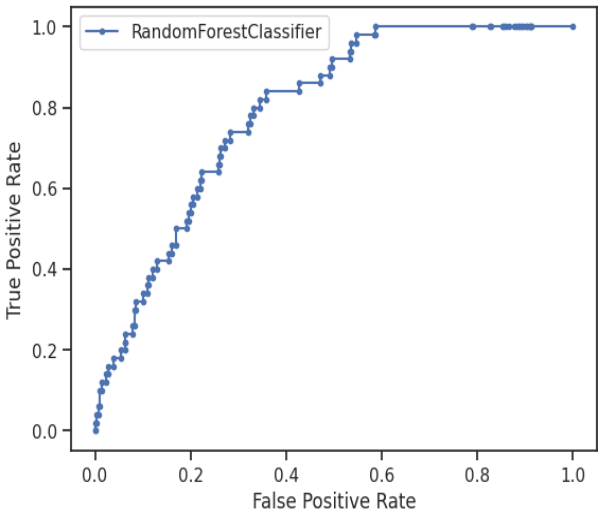
Selanjutnya merupakan hasil evaluasi dari algoritma random forest. dapat dilihat pada gambar 4 hasil evaluasi *metrics score* dari algoritma ini menghasilkan akurasi sebesar 87%, akurasi *k-fold cross validation* menghasilkan sebesar 92,4%. Selanjutnya, dihasilkan nilai *Precision, Recall*, dan *F1 score* secara berturut-urut sebesar 92%, 87%, 89%. Hasil dari penilaian *metrics score* dari algoritma ini tergolong sangat baik untuk proses klasifikasi. Untuk penjelasan lebih lengkap dapat dilihat pada gambar 4 dibawah ini.

```
RandomForestClassifier :
Accuracy Score: 0.8738

K-Fold Validation Mean Accuracy: 92.4660 %

Classification report:
              precision    recall  f1-score   support

           0       0.96      0.90      0.93       972
           1       0.14      0.32      0.20        50

    accuracy                           0.87      1022
   macro avg       0.55      0.61      0.57      1022
weighted avg       0.92      0.87      0.90      1022


Confusion matrix:
[[877  95]
 [ 34  16]]

Precision Score: 0.9226

Recall Score: 0.8738

ROC Score: 0.6111

F1 score: 0.8956
```

Gambar 6. *Metrics Score* Random Forest

Gambar diatas merupakan gambaran hasil penilaian *metrics score* dari algoritma random forest. Selanjutnya akan diperlihatkan hasil dari perhitungan TPR dan FPR dari algoritma ini yang akan direpresentasikan dalam bentuk kurva ROC, berikut merupakan hasil dari perhitungan TPR dan FPR dalam kurva ROC algoritma random forest.

p-ISSN: 2986-3929
e-ISSN: 3032-1948



Gambar 7. Kurva ROC Random Forest

c. Hasil Perbandingan Model

Setelah mengevaluasi kedua model, akan dilakukan perbandingan hasil *metrics score* dari kedua model. Hasil keseluruhan dari model yang digunakan dapat dilihat pada tabel berikut.

Tabel 2. *Metrics Score* Model

Model	Cross Val Acc	Test Acc	Precision	Recall	F1
Random Forest	0.924660	0.873777	0.922633	0.873777	0.895644
Decision Tree C4.5	0.887376	0.852250	0.920588	0.852250	0.882462

Berdasarkan Tabel diatas didapatkan bahwa algoritma Random Forest disertai *HyperParameter Tuning* mendapatkan performa yang lebih baik dari Algoritma C4.5, yaitu dengan akurasi *K-fold cross validation* sebesar 92,4%, *Precision* 92,2%, *Recall* 87,3%, dan 89,5% pada perhitungan *F1*.

3.2. Hasil Penggabungan Model

Setelah menyelesaikan semua langkah-langkah pengoptimalan performa setiap model, kedua model akan masuk ke tahap *emerging* atau penyatuan. Pada tahap ini, masing-masing model akan digabungkan menjadi model *ensemble* yang lebih kuat. Dengan memanfaatkan kekuatan masing-masing model, penyatuan berpotensi memberikan prediksi yang lebih akurat yang dapat diterapkan. Tujuan dari proses ini adalah untuk mendapatkan manfaat dari diversitas masing-masing model, sehingga meningkatkan kemampuan prediktif model setelah digabungkan. Berikut merupakan hasil dari penyatuan kedua model.

```
Accuracy Score: 0.8816

K-Fold Validation Mean Accuracy: 92.4660 %

Classification report:
              precision    recall  f1-score   support

           0       0.96      0.91      0.94       972
           1       0.15      0.30      0.20        50

    accuracy                           0.88      1022
   macro avg       0.56      0.61      0.57      1022
weighted avg       0.92      0.88      0.90      1022


Confusion matrix:
[[886  86]
 [ 35  15]]

Precision Score: 0.9222

Recall Score: 0.8816

F1 score: 0.9000
```

Gambar 8. Hasil *Emerging* Model

Setelah dilakukan proses penggabungan dari kedua model, didapatkan peningkatan hasil dari beberapa penilaian *metrics score*. Dapat dilihat pada gambar 6 diatas model baru yang dihasilkan memperoleh nilai akurasi dan akurasi *k-fold cross validation* secara berturut-urut yaitu sebesar 88% dan 92,4%. Selanjutnya, untuk niali *precision, recall*, dan *F1* score menghasilkan nilai sebesar 92%, 88%, dan 90%.

4. Kesimpulan

Berdasarkan dari penelitian yang telah dilakukan, algoritma C4.5 dan Random Forest merupakan algoritma yang sangat baik digunakan dalam proses klasifikasi penyakit stroke. Hal ini dibuktikan dengan semua hasil pengukuran *metrics score* dengan *HyperParameter Tuning* berada diatas 85%. Selain itu, dengan dilakukannya proses *emerging* kedua model ini dapat menghasilkan sebuah model baru yang lebih kuat dari kedua model sebelumnya. Model hasil emerging mendapatkan nilai *recall* terbaik sebesar 88%, selain itu, nilai *precision* dan F1 dari model ini menghasilkan nilai yang cukup baik yaitu sebesar 92%, dan 90%. Model hasil penggabungan algoritma C4.5 dan Random Forest terbukti dapat menghasilkan sebuah model baru yang lebih efisien dibandingkan dengan model algoritma C4.5 yang digunakan oleh Putri yang menghasilkan akurasi model sebesar 90%, sedangkan model hasil penggabungan pada penelitian ini dapat menghasilkan akurasi klasifikasi sebesar 92%. Pada penelitian selanjutnya diharapkan dapat mengurangi ketidakseimbangan dalam data, agar dapat diperoleh hasil *metrics score* yang lebih baik.

Daftar Pustaka

[1] A. U. Saputra, S. Mardiono, F. Kebidanan, D. Keperawatan, and E. Co, “Edukasi Kesehatan Tentang Perawatan Lansia Dengan Kejadian Stroke Di Rumah,” *Indonesian Journal Of Community Service*, vol. 2, 2022.

[2] Y. Ayuningtyas and I. Made Suartana, “Klasifikasi Penyakit Stroke Menggunakan Support Vector Machine (SVM) dan Particle Swarm Optimization (PSO),” *Journal of Informatics and Computer Science*, 2023.

[3] S. Ucha Putri, E. Irawan, F. Rizky, S. Tunas Bangsa, P. A. -Indonesia Jln Sudirman Blok No, and S. Utara, “Implementasi Data Mining Untuk Prediksi Penyakit Diabetes Dengan Algoritma C4.5,” 2021.

[4] O. Maha *et al.*, “Prediksi Paket Return Menggunakan Metode Decision Tree Menerapkan Algoritma C4.5 Berbasis Website.”

[5] G. Taufik and D. Jatmika, “Penerapan Algoritma C4.5 Untuk Klasifikasi Keberhasilan Pengiriman Barang,” vol. 6, no. 1, p. 2021.
[6] R. Chairunisa and W. Astuti, “Perbandingan CART dan Random Forest untuk Deteksi Kanker berbasis Klasifikasi Data Microarray,” *masa berlaku mulai*, vol. 1, no. 3, pp. 805–812, 2017.
[7] K. Dwitya Adhi Pradyto and M. A. Raharja, “Implementasi Random Forest dengan LASSO Dalam Klasifikasi Penyakit yang Ditularkan Melalui Nyamuk,” 2023. [Online]. Available: https://www.kaggle.com/datasets/richardbernat/vector-borne-disease-
[8] C. M. Sidiq, A. Faqih, and G. Dwilestari, “Algoritma Decision Tree C4.5 Digunakan Untuk Mengklasifikasikan Data Stroke,” 2024. [Online]. Available: www.kaggle.com,
[9] K. B. Lesmana *et al.*, “Deteksi Penyakit Diabetes Menggunakan Gaussian Naive Bayes, Regresi Logistik, dan Random Forest,” 2023. [Online]. Available: https://www.kaggle.com/datasets/akshaydattatraykhare/diabetes-dataset
[10] I. G. A. G. A. Kadyanan, “Sistem Pengenalan Kain Endek Khas Bali Berdasarkan Fitur Tekstur,” *Jurnal Ilmiah Ilmu Komputer Universitas Udayana*, 2016.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Halaman ini sengaja dibiarkan kosong