

PENERAPAN *MACHINE LEARNING* UNTUK PREDIKSI PENYAKIT STROKE

Denis Eka Cahyani^{1,*}

¹ Jurusan Matematika, FMIPA, Universitas Negeri Malang

Email: denis.eka.cahyani.fmipa@um.ac.id (D. E. Cahyani)

* Corresponding Author

Abstract

Stroke is a global health problem and one of the leading causes of adult disability. Early detection and prompt treatment are needed to minimize further damage to the affected brain area and complications to other parts of the body. Machine learning techniques can be used to predict stroke detection. Machine learning algorithms such as Naive Bayes, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Decision Tree are compared in this study to obtain the best performance in predicting stroke. The implementation stages in this research consist of the pre-processing data, the application of the algorithm and the evaluation and analysis. The Naïve Bayes algorithm obtains better Accuracy, Precision, Recall, and F1-Measure values compared to other algorithms. The values of Accuracy, Precision, Recall, and F1-Measure obtained by Naïve Bayes are 93.93%, 88.23%, 93.93%, and 91.00%, respectively. So the conclusion of this study is that the Naïve Bayes algorithm has the best performance compared to the SVM, KNN and Decision Tree algorithms in predicting stroke.

Keywords: *decision tree, klasifikasi, k-nearest neighbor, naïve bayes, stroke, support vector machine*

Submitted: 5 December 2021; Revised: 20 December 2021; Accepted Publication: 5 January 2022;

Published Online: January 2022

DOI: [10.17977/um055v3i1p15-22](https://doi.org/10.17977/um055v3i1p15-22)

PENDAHULUAN

Stroke merupakan masalah kesehatan global dan salah satu penyebab utama kecacatan orang dewasa. Menurut WHO, stroke merupakan penyakit paling mematikan nomor dua setelah penyakit jantung. Setiap tahunnya, lima belas juta orang menderita stroke di seluruh dunia dan penderita stroke meninggal setiap 4-5 menit (Campbell & Khatri, 2020). Negara berkembang seperti Indonesia perlu memperhatikan masalah penyakit stroke secara serius karena mempunyai angka kematian yang tinggi dan kasus yang terus meningkat (Hankey, 2017).

Stroke terjadi bila aliran darah ke berbagai area otak terganggu atau berkurang. Sel-sel di area otak tersebut tidak mendapatkan nutrisi dan oksigen sehingga mengakibatkan kematian sel. Stroke merupakan keadaan darurat medis yang membutuhkan perawatan segera. Deteksi awal dan penanganan yang tepat diperlukan untuk meminimalkan kerusakan lebih lanjut di area otak yang terkena dan komplikasi pada bagian tubuh lainnya (Ouyang & Sabat, 2020). Diagnosis awal stroke masih merupakan tantangan dalam bidang kedokteran. Teknik *machine learning* dapat digunakan dalam memprediksi penyakit stroke (Singh et al., 2020). *Machine learning* adalah metode analisis data yang dilakukan dengan otomatisasi pembuatan model analitis (Kanika & Sangeeta, 2019).

Beberapa penelitian sebelumnya telah menggunakan teknik *machine learning* untuk memprediksi stroke (Sailasya & Kumari, 2021; Emon et al., 2020; Shafiul Azam et al., 2020). Algoritma *machine learning* seperti Support Vector Machine (SVM) digunakan untuk memprediksi stroke dan menghasilkan akurasi sebesar 90% (Jeena & Kumar, 2017). Rajora dkk melakukan prediksi stroke menggunakan beberapa algoritma *machine learning* seperti Naïve Bayes, Logistic Regression, Gradient Boosting dan Random Forest (Rajora et al., 2021).

Selanjutnya, Sailasya dkk melakukan analisis performa prediksi stroke menggunakan beberapa algoritma *machine learning* dan menghasilkan akurasi tertinggi pada algoritma Naïve Bayes sebesar 82% (Sailasya & Kumari, 2021). Penelitian (Emon et al., 2020) menunjukkan bahwa model dilatih menggunakan algoritma Decision Tree, Random Forest, Multi-layer perceptron untuk memprediksi stroke. Akurasi yang diperoleh untuk Decision Tree adalah 74,31%, Random Forest sebesar 74,53%, dan Multi-layer perceptron sebesar 75,02%.

Keterbatasan pada penelitian sebelumnya adalah penelitian sebelumnya belum membandingkan kinerja algoritma *machine learning* yaitu K-Nearest Neighbor (KNN), Naive Bayes, Support Vector Machine (SVM), dan Decision Tree sekaligus. Oleh karena itu, penelitian ini berkontribusi untuk menerapkan algoritma *machine learning* yaitu KNN, Naïve Bayes, SVM dan Decision Tree untuk mendapatkan hasil kinerja terbaik. Sehingga penelitian ini dapat memprediksi penyakit stroke menggunakan algoritma *machine learning* dengan performa yang terbaik.

METODE

Metode terdiri dari tiga bagian, yaitu deskripsi data, prosedur implementasi, algoritma *machine learning* & matriks evaluasi. Ketiga proses tersebut adalah dijelaskan di bawah ini.

Deskripsi Data

Data pada penelitian ini merupakan dataset yang diambil dari laman (Fedesoriano, 2021). Dataset ini digunakan untuk memprediksi kemungkinan pasien terkena stroke berdasarkan parameter input seperti jenis kelamin, usia, penyakit yang diderita, dan status merokok. Dataset tersebut berjumlah 5110 data dan 12 atribut. Atribut dalam dataset tersebut ditunjukkan pada Tabel 1.

Tabel 1. Atribut Dataset Stroke (Fedesoriano, 2021)

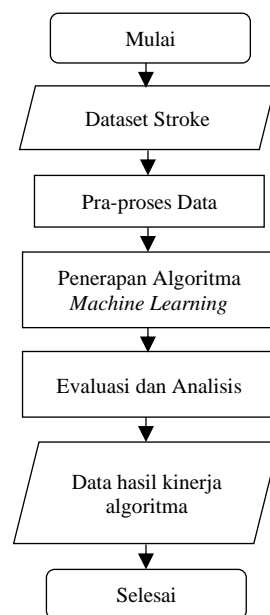
Nama Atribut	Deskripsi	Tipe
Id	Nilai bilangan bulat unik pasien	Data numerik
Usia	Usia seseorang	Data numerik
Jenis Kelamin	Jenis kelamin seseorang	Data kategorik
Hipertensi	Seseorang mempunyai hipertensi atau tidak	Data numerik
Jenis pekerjaan	Jenis pekerjaan seseorang	Data kategorik
Tipe tempat tinggal	Tipe tempat tinggal seseorang	Data kategorik
Riwayat penyakit jantung	Seseorang mempunyai riwayat sakit jantung atau tidak	Data numerik
Tingkat glukosa rata-rata	Tingkat kondisi glukosa seseorang	Data numerik
BMI	Indeks massa tubuh seseorang	Data numerik
Status menikah	Status menikah seseorang	Data kategorik
Status merokok	Status kondisi merokok seseorang	Data kategorik
Stroke	Target output seseorang stroke atau tidak	Data numerik

Prosedur Implementasi

Prosedur implementasi pada penelitian ini terdiri dari beberapa tahap, yaitu pra-proses data, penerapan algoritma *machine learning*, dan evaluasi dan analisis. Prosedur implementasi penelitian ditunjukkan pada Gambar 1. Prosedur penelitian dimulai dari pengambilan dataset stroke dari laman (Fedesoriano, 2021). Selanjutnya proses dilanjutkan dengan pra-proses data berupa pembersihan data dengan menghilangkan noise dan outlier pada dataset. Pada pra-proses juga dilakukan pengkodean label yang mengkodekan data kategorik menjadi data numerik. Pada dataset terdapat lima atribut berupa data kategorik yang memiliki tipe data string.

Pengkodean label ini mengubah semua data bertipe string menjadi integer/angka. Tahap selanjutnya yaitu penerapan algoritma *machine learning*. Sebelum diterapkan, data sebelumnya dibagi menjadi data latih dan uji untuk akurasi dan efisiensi yang lebih baik. Rasio data latih dan data uji yaitu 80%:20%. Selanjutnya, berbagai algoritma *machine learning* digunakan untuk melatih model. Algoritma klasifikasi yang digunakan pada penelitian ini adalah Naïve Bayes, Support Vector Machine, K-Nearest Neighbors, dan Decision Tree. Selanjutnya, tahap evaluasi dilakukan untuk mengevaluasi hasil klasifikasi dataset menggunakan confusion matrix. Confusion matrix digunakan untuk mengetahui *accuracy*, *precision*, *recall*, dan *F1-measure*. Hasil confusion matrix selanjutnya dianalisis untuk mendapatkan kinerja model algoritma yang optimal. Terakhir diperoleh data hasil kinerja algoritma yang sudah diterapkan dan diketahui algoritma yang paling optimal untuk melakukan prediksi penyakit stroke.

Pada penelitian ini menggunakan hardware untuk penerapan algoritma *machine learning* menggunakan hardware sebagai berikut Motherboard, Processor Core i7, RAM (Random Access Memory) 16 GB kecepatan 3600MHz. Sedangkan software yang dibutuhkan sebagai berikut Windows 10, python 3.10, dan jupyter notebook.



Gambar 1. Prosedur Implementasi Penelitian

Algoritma Machine Learning

Algoritma *machine learning* yang diterapkan dalam penelitian ini adalah Naïve Bayes, SVM, KNN, dan Decision Tree. Berikut deskripsi masing-masing algoritma.

Naïve Bayes

Naive Bayes merupakan salah satu metode statistik probabilitas berdasarkan penerapan *Teorema Bayes* dengan asumsi independen (naif) yang kuat, untuk memprediksi kelas dari suatu dokumen berdasarkan probabilitasnya (Wisudawati et al., 2017).

Teorema Bayes dinyatakan dengan rumus:

$$P(C|X) = \frac{P(X|C) \times P(C)}{P(X)} \quad (1)$$

Keterangan:

- X : Data dengan kelas yang belum diketahui
- C : Hipotesis data merupakan suatu kelas spesifik
- $P(C|X)$: Probabilitas hipotesis berdasarkan kondisi (*posteriori probability*)
- $P(C)$: Probabilitas hipotesis (*prior probability*)
- $P(X|C)$: Probabilitas berdasarkan kondisi pada hipotesis

$P(X)$: Probabilitas hipotesis X

Rumus diatas dapat ditulis sebagai berikut:

$$\text{Posterior probability} = \frac{\text{conditional probability} \times \text{prior probability}}{\text{evidence}} \quad (2)$$

Keterangan:

Posterior probability : Peluang masuknya sampel karakteristik tertentu pada suatu kelas

Prior probability : Peluang munculnya suatu kelas

Conditional probability : Peluang kemunculan karakteristik-karakteristik sampel pada kelas A

Evidence : Peluang kemunculan karakteristik sampel secara global
maka berlaku suatu persamaan sebagai berikut:

$$P(C|X_1, \dots, X_n) = \prod_{i=1}^n P(X_i|C) \times P(C) \quad (3)$$

Support Vector Machine (SVM)

Konsep klasifikasi SVM yaitu berusaha untuk membuat sebuah garis pemisah (*hyperlane*) yang ideal dalam ruang komponen dimensi yang lebih tinggi untuk memetakan informasi dengan meminimalkan resiko (Kumar, Virmani, Bhadauria, & Panda, 2018). *Hyperlane* dibangun dengan menggunakan *support vector*, data yang lebih dekat dengan *hyperlane*. Data ini terletak pada batas irisan dengan kelas pertama disebut *support vector +* (positif), dan kelas kedua *support vector -* (negative). Jarak antara *support vector* disebut margin, dimana maximum margin menandakan *hyperlane* yang baik. Persamaan garis *hyperlane* adalah

$$g(x) = w^T x + b \quad (4)$$

Keterangan:

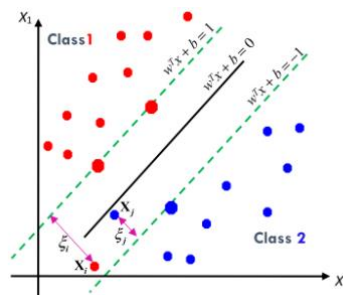
w^T : Vector dimensi n

b : Bias term

Pada data terdekat dengan *hyperlane* ditarik garis sebagai *support vector* (Gambar 2), dimana

$$g(x) \geq 1; \text{ kelas 1} \quad (5)$$

$$g(x) \leq -1; \text{ kelas 2} \quad (6)$$



Gambar 2. Ilustrasi SVM (Gholami & Fakhari, 2017)

Tujuan dari SVM adalah mencari *hyperlane* yang paling optimal untuk membagi data agar benar-benar terpisah menjadi 2 bagian. Lalu bagaimana bila data yang ada memang tidak bisa dipisahkan secara linear, Gholami & Fakhari (2017) mengatakan bahwa untuk memecahkan permasalahan ini dapat juga memetakan data input ke dimensi ruang *dot product* yang lebih tinggi yang dikenal sebagai *feature space* atau *Hilbert space*. Maka dapat dikatakan bahwa pada input *space* data tetaplah *nonlinear*, namun dapat diubah ke *feature space* dan dicari *hyperplane*-nya. Karena sulitnya menemukan fungsi transformasi $\phi(x_i)$ yang tepat untuk sebuah data, maka dapat menggunakan fungsi kernel yang memenuhi teorema Mercer bahwa matriks kernel yang dihasilkan harus bersifat positif semi-definite. Fungsi kernel yang biasa digunakan adalah Linear, Polynomial, sigmoid, dan Radial Basis Function (RBF) (Gholami & Fakhari, 2017).

K-Nearest Neighbor (KNN)

K-Nearest Neighbor adalah metode *supervised learning* dimana tujuan dari metode ini adalah melakukan klasifikasi obyek baru berdasarkan atribut dan data latih. Untuk melakukan klasifikasi *KNN*, dilakukan pembobotan terlebih dahulu kemudian dilakukan perankingan dokumen berdasarkan tingkat kesesuaian dokumen terhadap *query*, kemudian *KNN* bekerja berdasarkan jarak minimum dari data baru ke data *training samples* untuk menentukan K tetangga terdekat (Cunningham & Delany, 2021). Berikut persamaan *KNN* yang digunakan untuk menghitung jarak antara data baru dan semua data yang ada di data *training* dengan menggunakan *Euclidean Distance* dengan rumus (Mesquita et al., 2017):

$$D(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2} \quad (7)$$

Keterangan:

x : Nilai term x pada dokumen i
 y : Nilai term y pada dokumen i
 D : Distance (jarak) Euclidean
 d : Jumlah variabel

Decision Tree

Decision Tree merupakan algoritma *machine learning* yang membangun klasifikasi atau regresi model dalam bentuk struktur pohon. Hal tersebut dilakukan dengan cara memecah terus ke dalam himpunan bagian yang lebih kecil lalu pada saat itu juga sebuah pohon keputusan secara bertahap dikembangkan. Hasil akhir dari proses tersebut adalah pohon dengan node keputusan dan node daun (Song & Lu, 2015).

Algoritma Decision Tree dimulai dengan membuat cabang untuk setiap nilai, selanjutnya memilih atribut sebagai root, kemudian ulangi prosedur untuk setiap cabang sampai semua kasus di cabang memiliki kelas yang sama. Terakhir memilih atribut berdasarkan nilai gain tertinggi dari atribut-atribut yang ada. Perhitungan gain menggunakan Rumus 8 dan perhitungan nilai entropy menggunakan Rumus 9 (Bulac & Bulac, 2016).

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n * Entropy(S_i) \quad (8)$$

Keterangan:

S : Himpunan
 A : Atribut
 n : Jumlah partisi atribut A
 $/ S_i /$: Jumlah kasus pada partisi ke- i
 $/ S /$: Jumlah kasus dalam S

$$Entropy(S) = \sum_{i=1}^n - p_i * \log_2 p_i \quad (9)$$

Keterangan:

S : Himpunan kasus
 A : Fitur
 n : Jumlah partisi S
 p_i : Proporsi dari S_i terhadap S

Matriks Evaluasi

Matriks evaluasi pada penelitian ini digunakan untuk mengukur kinerja suatu metode klasifikasi diperlukan untuk mengetahui seberapa baik sistem dalam melakukan klasifikasi data. Pengujian dilakukan dengan menggunakan *confusion matrix* dimana pengujian dan evaluasi hasil pengukuran kualitas klasifikasi dengan membangun *confusion matrix* mencatat contoh yang dikenali secara benar dan salah oleh masing-masing kelas (Sokolova & Lapalme, 2009). Ada empat nilai yang dihasilkan di dalam tabel *confusion matrix*, diantaranya *True*

Positive (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN). Ilustrasi tabel confusion matrix dapat dilihat pada Gambar 2. Pada tahap evaluasi dilakukan perhitungan *Accuracy*, *Precision*, *Recall*, and *F1-Measure* yang ditunjukkan pada Rumus (10), (11), (12), and (13).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

$$Recall = \frac{TP}{TP+FN} \quad (11)$$

$$Precision = \frac{TP}{TP+FP} \quad (12)$$

$$F1 - Measure = 2 * \frac{(Precision*Recall)}{(Precision+Recall)} \quad (13)$$

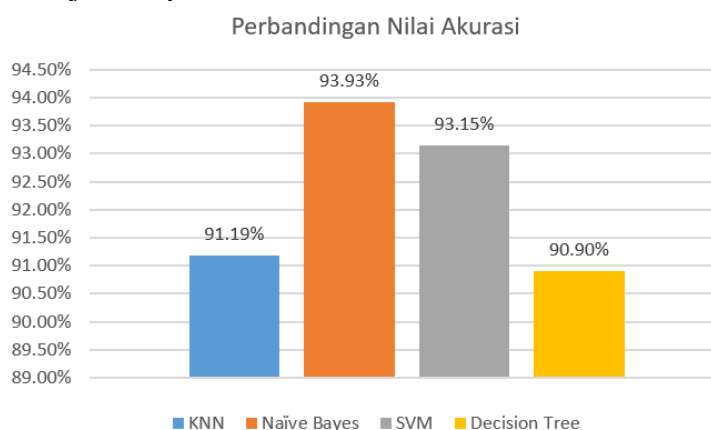
Nilai Aktual

Nilai Prediksi		Positive	Negative
	Positive	TP	FP
	Negative	FN	TN

Gambar 2. Ilustrasi Tabel Confussion Matrix

HASIL DAN PEMBAHASAN

Penelitian ini membandingkan kinerja algoritma *machine learning* yaitu Naive Bayes, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), dan Decision Tree untuk memprediksi penyakit stroke. Dari algoritma yang digunakan, Naive Bayes mempunyai kinerja yang paling baik dengan akurasi 93,93%. Sedangkan algoritma lainnya yaitu Support Vector Machine (SVM), K-Nearest Neighbor (KNN), dan Decision Tree menghasilkan akurasi berturut-turut sebesar 93,15%, 91,19%, dan 90,90%. Perbandingan akurasi yang diperoleh dari algoritma tersebut ditunjukkan pada Gambar 3.



Gambar 3. Perbandingan Nilai Akurasi

Evaluasi prediksi penyakit stroke pada penelitian ini juga menggunakan *precision*, *recall*, dan *F1-measure*. Tabel 2 menunjukkan perbandingan nilai *precision*, *recall*, dan *F1-measure* dari algoritma Naive Bayes, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), dan Decision Tree. Algoritma Naive Bayes memiliki nilai *precision*, *recall*, dan *F1-measure* yang relatif lebih tinggi dibandingkan dengan algoritma lainnya. Nilai *precision*, *recall*, dan *F1-measure* pada Naive Bayes berturut-turut adalah 88,23%, 93,93%, dan 91,00%. Perbandingan *precision*, *recall*, dan *F1-measure* ditunjukkan pada Tabel 2.

Tabel 2. Perbandingan nilai precision, recall, and F1-measure

	Precision	Recall	F1-Measure
Naïve Bayes	88,23%	93,93%	91,00%
SVM	88,88%	93,15%	90,77%
KNN	89,29%	91,19%	90,19%
Decision Tree	89,93%	90,90%	90,40%

Pada penelitian ini algoritma Naïve Bayes memperoleh hasil evaluasi yang paling baik dibandingkan dengan algoritma lainnya yaitu SVM, KNN, dan Decision Tree. Hal ini sesuai dengan penelitian terdahulu (Xhemali et al., 2009) yang mengatakan bahwa algoritma Naïve Bayes memiliki tingkat akurasi yg lebih baik dibanding algoritma klasifikasi lainnya. Algoritma ini hanya membutuhkan jumlah data latih yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian sehingga memperoleh hasil klasifikasi terbaik. Sedangkan untuk algoritma *machine learning* lainnya seperti SVM, KNN dan Decision Tree menghasilkan akurasi yang lebih rendah dikarenakan algoritma tersebut tidak efektif jika jumlah data latih sedikit.

Kelebihan dari algoritma naïve bayes juga mengasumsikan antar variabel sebagai variabel independent sehingga hanya varians dari suatu variabel dalam sebuah kelas yang dibutuhkan untuk menentukan klasifikasi, bukan keseluruhan dari matriks kovarians. Algoritma Naïve Bayes juga sangat baik untuk digunakan dalam klasifikasi data biner seperti dataset yang digunakan pada penelitian ini yang mengklasifikasi pasien menjadi penderita stroke atau bukan. Sehingga algoritma Naïve Bayes sangat efektif bekerja dengan dataset stroke yang digunakan dalam penelitian ini.

PENUTUP

Penelitian ini memprediksi penyakit stroke dengan membandingkan kinerja algoritma *machine learning* yaitu Naive Bayes, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), dan Decision Tree untuk mendapatkan hasil kinerja terbaik. Algoritma Naïve Bayes memperoleh nilai *Accuracy*, *Precision*, *Recall*, dan *F1-Measure* yang lebih baik dibandingkan dengan algoritma lainnya. Nilai *Accuracy*, *Precision*, *Recall*, dan *F1-Measure* yang diperoleh Naïve Bayes secara berturut-turut sebesar 93,93%, 88,23%, 93,93%, dan 91,00%. Secara umum algoritma klasifikasi yang diterapkan dalam penelitian ini dapat memprediksi penyakit stroke dengan baik.

Penelitian selanjutnya dapat dikembangkan pada penelitian ini dengan menerapkan algoritma *machine learning* lainnya seperti Random Forest, Adaboost dan lainnya. Algoritma yang digunakan juga dapat diperluas dengan menggunakan Neural Network atau algoritma Deep Learning. Penelitian ini juga terbatas pada dataset berupa teks dan angka, yang tidak selalu akurat untuk memprediksi stroke. Dataset yang digunakan dapat diperluas dengan dataset yang terdiri dari gambar seperti CT Scan otak untuk memprediksi kemungkinan stroke yang lebih efisien kedepannya.

DAFTAR RUJUKAN

- Bulac, C., & Bulac, A. (2016). Decision Trees. In *Advanced Solutions in Power Systems: HVDC, FACTS, and AI Techniques*. <https://doi.org/10.1002/9781119175391.ch18>
- Campbell, B. C. V., & Khatri, P. (2020). Stroke. In *The Lancet* (Vol. 396, Issue 10244). [https://doi.org/10.1016/S0140-6736\(20\)31179-X](https://doi.org/10.1016/S0140-6736(20)31179-X)
- Cunningham, P., & Delany, S. J. (2021). K-Nearest Neighbour Classifiers-A Tutorial. In *ACM Computing Surveys* (Vol. 54, Issue 6). <https://doi.org/10.1145/3459665>
- Emon, M. U., Keya, M. S., Meghla, T. I., Rahman, M. M., Mamun, M. S. al, & Kaiser, M. S. (2020). Performance Analysis of Machine Learning Approaches in Stroke Prediction.

- Proceedings of the 4th International Conference on Electronics, Communication and Aerospace Technology, ICECA* 2020.
<https://doi.org/10.1109/ICECA49313.2020.9297525>
- FEDESORIANO. (2021). *Stroke Prediction Dataset*.
<https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset>
- Gholami, R., & Fakhari, N. (2017). SUPPORT VECTOR MACHINE : In *Handbook of Neural Computation* (1st ed.). Elsevier Inc. <https://doi.org/10.1016/B978-0-12-811318-9.00027-2>
- Hankey, G. J. (2017). Stroke. In *The Lancet* (Vol. 389, Issue 10069).
[https://doi.org/10.1016/S0140-6736\(16\)30962-X](https://doi.org/10.1016/S0140-6736(16)30962-X)
- Jeena, R. S., & Kumar, S. (2017). Stroke prediction using SVM. *2016 International Conference on Control Instrumentation Communication and Computational Technologies, ICCICCT 2016*. <https://doi.org/10.1109/ICCICCT.2016.7988020>
- Kanika, & Sangeeta. (2019). Applying machine learning algorithms for news articles categorization: Using SVM and kNN with TF-IDF approach. In *Smart Computational Strategies: Theoretical and Practical Aspects*. https://doi.org/10.1007/978-981-13-6295-8_9
- Kumar, I., Virmani, J., Bhadauria, H. S., & Panda, M. K. (2018). Classification of Breast Density Patterns Using PNN, NFC, and SVM Classifiers. In *Soft Computing Based Medical Image Analysis*. Elsevier Inc. <https://doi.org/10.1016/B978-0-12-813087-2.00012-9>
- Mesquita, D. P. P., Gomes, J. P. P., Souza Junior, A. H., & Nobre, J. S. (2017). Euclidean distance estimation in incomplete datasets. *Neurocomputing*, 248.
<https://doi.org/10.1016/j.neucom.2016.12.081>
- Ouyang, T., & Sabat, S. (2020). Stroke. In *Radiology Fundamentals: Introduction to Imaging & Technology*. https://doi.org/10.1007/978-3-030-22173-7_54
- Rajora, M., Rathod, M., & Naik, N. S. (2021). Stroke prediction using machine learning in a distributed environment. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12582 LNCS. https://doi.org/10.1007/978-3-030-65621-8_15
- Sailasya, G., & Kumari, G. L. A. (2021). Analyzing the Performance of Stroke Prediction using ML Classification Algorithms. *International Journal of Advanced Computer Science and Applications*, 12(6). <https://doi.org/10.14569/IJACSA.2021.0120662>
- Shafiul Azam, Md., Habibullah, Md., & Kabir Rana, H. (2020). Performance Analysis of Various Machine Learning Approaches in Stroke Prediction. *International Journal of Computer Applications*, 175(21). <https://doi.org/10.5120/ijca2020920740>
- Singh, M. S., Choudhary, P., & Thongam, K. (2020). A comparative analysis for various stroke prediction techniques. *Communications in Computer and Information Science*, 1148 CCIS. https://doi.org/10.1007/978-981-15-4018-9_9
- Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*.
<https://doi.org/10.1016/j.ipm.2009.03.002>
- Song, Y. Y., & Lu, Y. (2015). Decision tree methods: applications for classification and prediction. *Shanghai Archives of Psychiatry*, 27(2). <https://doi.org/10.11919/j.issn.1002-0829.215044>
- Wisudawati, J., Adiwijaya, & Faraby, S. al. (2017). Klasifikasi Sentimen pada Movie Review dengan Metode Multinomial Naïve Bayes. *E-Proceeding of Engineering*, 4(2).
- Xhemali, D., J. Hinde, C., & G. Stone, R. (2009). Naive Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages. *International Journal of Computer Science*, 4(1).