Bitlis Eren Üniversitesi Fen Bilimleri Dergisi
BİTLİS EREN UNIVERSITY JOURNAL OF SCIENCE
ISSN: 2147-3129/e-ISSN: 2147-3188
VOLUME: 13 NO: 4 PAGE: 1169-1180 YEAR: 2024
DOI: 10.17798/bitlisfen.1539189

Detection of Stroke (Cerebrovascular Accident) Using Machine Learning Methods

Hadice ATEŞ[1], Abidin ÇALIŞKAN[2*]

[1]Batman University, Department of Electrical and Electronics Engineering, Batman, Türkiye
[2]Batman University, Department of Computer Engineering, Batman, Türkiye
(ORCID: 0000-0001-5039-6400) (ORCID: 0000-0002-0462-6997)

Keywords: Stroke, Artificial intelligence, Machine Learning, Classification, Accuracy.

Abstract

Stroke occurs when the blood flow to the brain is suddenly interrupted. This interruption can lead to the loss of function in the affected area of the brain and cause permanent damage to the corresponding part of the body. Stroke can develop due to various factors such as age, occupation, chronic diseases, and a family history of stroke. Assessing these factors and predicting stroke risk is often a costly and time-consuming process, which can increase the risk of permanent damage for the individual. However, with today's technology, Artificial Intelligence (AI) and Machine Learning (ML) models can process millions of data points to determine stroke risk within seconds. In this study, the risk of stroke in individuals is predicted most reliably using ML methods such as Logistic Regression (LR), Decision Tree (DT), Support Vector Machines (SVM), and k-Nearest Neighbors (KNN), with the aim of saving time, protecting human health, and enabling early diagnosis of the disease. As a result of the study, the highest accuracy rate was achieved by the DT model with 91%. The accuracy rates of the other models were found to be 89% for SVM, 81% for KNN, and 75% for LR.

1. Introduction

Stroke, medically referred to as “stroke” or “paralysis,” is a condition that leads to the loss of muscle movement in a specific area of the body or the entire body [1]. It typically occurs due to an interruption or reduction in blood flow to the brain. This interruption causes brain cells to be damaged due to a lack of oxygen and nutrients [1], [2]. Depending on the region of the brain affected by the stroke, it can lead to various sensory and motor function losses [2]. Stroke is categorized into three main types:

Ischemic stroke occurs as a result of a blockage in an artery that supplies blood to the brain and is the most common type of stroke. It usually occurs due to the blockage of blood vessels by causes such as blood clots or atherosclerosis.

Hemorrhagic stroke happens when a blood vessel in the brain ruptures and blood leaks into the brain tissue. This type of stroke is associated with conditions such as high blood pressure, aneurysms, or head trauma.

A transient ischemic attack is a temporary condition caused by a short-term reduction in blood flow to the brain. It typically lasts for a few minutes and does not cause permanent damage but can be a warning sign of a more severe stroke [3], [4].

The symptoms of a stroke appear suddenly and usually include sudden numbness and weakness in the face, arm, or leg, sudden confusion, difficulty speaking or understanding speech, sudden loss of vision or blurred vision, sudden difficulty walking, dizziness, loss of balance or coordination, and sudden severe headache. The presence of any of these symptoms requires urgent medical intervention [5].

Several factors increase the risk of stroke, including high blood pressure, diabetes, high cholesterol, smoking, obesity, physical inactivity, a family history of stroke, and heart disease.

[*]Corresponding author: *abidin.caliskan@batman.edu.tr*

Received: 27.08.2024, Accepted: 08.09.2024

Understanding these risk factors is crucial for stroke prevention and early intervention [6].

Stroke treatment varies depending on the type and severity of the stroke. In the case of an ischemic stroke, medications or surgical interventions may be used to dissolve or remove the clot. For hemorrhagic stroke, the goal is to stop the bleeding and reduce the pressure on the brain. The rehabilitation process includes methods such as physical therapy, speech therapy, and occupational therapy. Early diagnosis and treatment can significantly reduce the effects of stroke and increase the chances of recovery [6], [7].

The diagnosis of stroke is made through physical examination and clinical evaluation. The physician evaluates the patient's symptoms, medical history, and when the stroke symptoms began. This includes checking consciousness, speech ability, mobility, balance, and coordination. Additionally, the “FAST” test is used for rapid diagnosis; this test focuses on factors such as the face, arms, speech, and time [8].

AI and ML are two significant concepts that have a profound impact on the modern technology world. AI is a field of science aimed at enabling computers and machines to exhibit human-like intelligence behaviors. These behaviors include abilities such as problem-solving, learning, reasoning, and decision-making. ML, on the other hand, is a subfield of AI that allows computers to automatically perform specific tasks by learning from data. ML enables systems to improve their performance through experiences and examples without being explicitly programmed [9].

AI and ML are used in many fields. These technologies have created a significant revolution in the healthcare sector. They are particularly used in areas such as early diagnosis of diseases, treatment planning, drug discovery, and patient monitoring. AI-based systems can analyze large datasets to detect disease symptoms more quickly and accurately, contributing to the development of more effective treatment processes. ML models, trained on millions of medical data points, can successfully perform complex tasks such as predicting stroke risk or forecasting disease progression [10].

The use of AI and ML in the healthcare sector also makes it possible to develop clinical decision support systems. These systems assist doctors in making more accurate diagnoses and treatment decisions by providing better insights into patients' conditions. Additionally, with these technologies, patient data can be analyzed in real-time, enabling the early identification and prevention of potential complications. By making healthcare services more efficient, accessible, and personalized, AI and ML offer significant conveniences in the field of healthcare [11], [12].

2. Literature Review

Several studies have been conducted on stroke prediction datasets in the literature, utilizing various AI models.

Emon et al. (2020) evaluated 10 different ML methods for stroke prediction and found that the Weighted Voting Algorithm provided the highest performance with an accuracy rate of 97% [13].

Singh and Choudhary predicted stroke risk using AI techniques, achieving an accuracy rate of 97.7% by using the CART algorithm for feature selection, PCA for dimensionality reduction, and Backpropagation Neural Network (BP) for classification [14].

Sevli identified stroke risk using the RFC technique and achieved an accuracy rate of 98.84% by applying resampling to solve data problems; the study found that age, body mass index, and glucose level were significant in risk prediction [15].

Revanth et al. (2020) compared SVM, CART, RFC, and MLP models for stroke risk prediction and reported that SVM demonstrated the best performance with an accuracy rate of 98.99% [16].

Cheon et al. (2019) applied DO techniques using a large patient dataset and achieved an AUC value of 83.48% by performing dimensionality reduction with PCA, indicating relatively high model accuracy [17].

Shoily et al. (2019) compared the Naive Bayes, J48, k-Nearest Neighbor (KNN), and Random Forest algorithms for stroke detection and found that Naive Bayes performed worse with an accuracy rate of 85.6%, while J48, KNN, and Random Forest algorithms produced better results with an accuracy rate of 99.8% [18].

Pradeepa et al. (2020) applied spectral clustering and various ML algorithms to identify stroke symptoms using social media data, showing that PNN provided the highest performance with an accuracy rate of 89.90% [19].

Li et al. (2019) used LR, Naive Bayes, Bayesian Network, DT, Neural Network, Random Forest, and other algorithms to classify stroke risk levels, reporting that the boosting model created with DT achieved the best performance with a recall rate of 99.94%, while Random Forest stood out with a precision rate of 97.33% [20].

In Table 1, the performance values of the study conducted and literature studies have been compared.

Table 1. Performance Comparisons in the Literature

Authors	Year of Study	Dataset Used	Methods	Accuracy Rate
Proposed study	**2024**	**Stroke Prediction Dataset**	**LR** **KNN** **DVM** **DT**	**%75** **%89** **%81** **%91**
Emon and Choudhary	2020	Stroke Prediction Dataset	Weighted Voting Algorithm	%97
Singh et al.	2017	Cardiovascular Health Study (CHS) dataset	CART algorithm, Dimensionality reduction with PCA, Backpropagation Neural Network (BP)	%97
Sevli	2021	Stroke Prediction Dataset	RFC technique, resampling	%98
Revanth et al.	2020	Stroke Prediction Dataset	DVM, CART, RFC, MLP models	%98
Cheon et al.	2019	China National Stroke Screening Data	PCA	%83
Shoily et al.	2019	Stroke Prediction Dataset	Naive Bayes, J48, KNN, Random Forest	%85
Pradeepa et al.	2020	Social Media Content	Various ML algorithms	%89
Li et al.	2019	China National Stroke Screening Data	LR, DT, Naive Bayes	%97

3. Materials and Methods

3.1. Lojistic Regression

LR is a statistical analysis method primarily used to establish relationships between two or more categorical outcomes [21]. This model is widely preferred in cases where the dependent variable is binary (e.g., “yes” or “no”) or multi-class. Unlike linear regression, LR predicts the probabilities of outcomes and uses a logistic function that limits these probabilities between 0 and 1 [22].

In the LR model, the effect of independent variables (input variables) on the dependent variable (output variable) is examined. This effect is used to predict the likelihood of an event occurring (e.g., the probability of a patient contracting a particular disease) [23]. The model's fundamental assumption is that the probability distribution of the dependent variable can be well modeled by a logistic function. This function expresses the probability of the dependent variable as a linear combination of the independent variables [24].

LR is applied in many fields, including medical diagnosis, marketing analytics, social sciences, and credit risk assessment. The model allows for both understanding the impact of independent variables and producing useful results in classification problems [25].

This model is particularly powerful in classification problems and can yield highly accurate results when the structure of the data and the nature of the problem are suitable [26].

3.2. K-Nearest Neighbors

The working principle of the KNN algorithm is quite simple: when a new data point needs to be classified, the distance between this point and the other points in the dataset is calculated [27]. These distances are usually measured using metrics such as Euclidean distance or Manhattan distance. As a result of these calculated distances, the nearest K neighbors are selected, and the majority class of these neighbors is assigned as the predicted class for the new data point. The value of K is a parameter that determines the number of neighbors and directly affects the model's performance [28].

The KNN algorithm does not perform any complex calculations during model training, which classifies it as a “lazy learning” algorithm [29]. Although this feature means that KNN can be slow for large datasets, it can produce highly effective results for small and medium-sized datasets [30].

KNN is widely used in fields such as medicine, bioinformatics, marketing analytics, and recommendation systems. The simplicity and interpretability of the algorithm make it an attractive choice for data scientists and analysts.

However, the performance of the KNN algorithm depends on the chosen value of K, the distance metric used, and the characteristics of the dataset [31]. Since KNN relies on the proximity relationships of data points, it is important to properly scale the data and carefully select the features [32].

3.3. Support Vector Machines

SVM is a supervised learning method and a powerful algorithm used to solve classification and regression problems. SVM is particularly known as a flexible and robust classification technique that yields effective results in high-dimensional datasets [33].

The fundamental principle of SVM is to find a hyperplane (or decision boundary) that best separates the data. This hyperplane aims to separate data points from different classes with the maximum margin. The margin refers to the distance between the hyperplane and the nearest data points. By finding the hyperplane that maximizes this margin, SVM ensures accurate classification of the data [33], [34].

If the data is linearly separable, SVM directly finds this hyperplane. However, if the data is not linearly separable, SVM uses kernel functions to transform the data into a higher-dimensional space. This transformation allows the data to become linearly separable, enabling SVM to perform effective classification even on complex datasets. Commonly used kernel functions include linear, polynomial, radial basis function (RBF), and sigmoid functions [35].

One of the advantages of SVM is its ability to generalize well on large datasets. Additionally, its resistance to overfitting makes SVM a reliable model. However, the computational cost of SVM can be high, especially for large and complex datasets. Moreover, selecting optimal hyperparameters (e.g., the kernel function and regularization parameter) directly affects the model's performance and requires careful tuning [36], [37].

SVM is widely used in various fields such as bioinformatics, image processing, text classification, and genetic analysis. The high accuracy rates it demonstrates in classification tasks make SVM a significant tool in data science and ML projects [38].

3.4. Decision Trees

DT is a popular ML algorithm used to solve both classification and regression problems. DTs are employed to make decisions based on certain features of the data and to visualize these decisions in a simple and understandable way. This algorithm creates a model that resembles a tree structure; this model starts from the root node and branches out into branches and leaves [39].

DTs classify or predict data points based on the independent variables (features) in the dataset. Each node represents a feature, and the best splitting point for this feature is selected. This split is defined as a condition that divides the dataset into subgroups. The branching process continues until the data is fully separated or further splitting is no longer meaningful. Ultimately, the leaf nodes represent a specific class or value [40].

DTs work by creating clear decision rules on the data. For example, in a medical diagnosis system, DTs can be used to predict which disease a patient may have based on their age, symptoms, and test results. The tree evaluates the probabilities for each disease based on these factors and makes its final decision [41].

One of the biggest advantages of DTs is that the models they create are explainable and can be visualized. This allows for an easy understanding and interpretation of the model's decision-making process. Additionally, DTs can work with both numerical and categorical data and have the ability to handle missing data [42].

However, DTs also have some disadvantages. For example, an overgrown decision tree can negatively impact the model's performance. Pruning techniques can be applied to prevent this situation. Moreover, DTs may not be able to model the relationship between variables in a complex and non-linear way [43].

DTs have a wide range of applications in finance, marketing, medicine, biology, and many other fields. Furthermore, they form the basis for more complex algorithms such as Random Forests and Gradient Boosting. These algorithms combine multiple decision trees to create more robust and generalized models [44], [45].

3.5. Model Performance Metrics

Various measurement criteria are used to evaluate the performance of models used in ML methods. Among these criteria, accuracy, precision, sensitivity, and F1 score are the most common. It is important to calculate these metrics to objectively assess the success of a model. Each of these metrics reflects different performance characteristics.

The accuracy metric, commonly used to evaluate algorithm performance, represents the ratio of correctly classified examples to the total predictions made by the model. This metric encompasses both true positive and true negative predictions and is calculated as shown in formula (1) [46], [47].

$$Accuracy = \frac{TP+TN}{TP+TN+TN+FN} \quad \ldots\ldots\ldots(1)$$

Precision refers to the proportion of truly positive examples among the instances that the model has classified as positive. This metric indicates the accuracy of the predictions for the positive class and is explained by formula (2) [48], [49].

$$Precision = \frac{TP}{TP+TF} \quad \ldots\ldots\ldots(2)$$

Sensitivity (or recall) shows how accurately the model identifies true positive examples. This metric determines how many of the positive class examples are correctly classified as positive and is calculated as shown in formula (3) [50], [51].

$$Recall = \frac{TP}{TP+FN} \quad \ldots\ldots\ldots(3)$$

The F1 score provides a balanced assessment of precision and sensitivity by taking their harmonic mean. It is used to evaluate the overall performance of the model, particularly in cases of class imbalance, and the calculation method is shown in formula (4) [52], [53].

$$F1\ Score = 2 * \frac{precision*recall}{precision+recall} \quad \ldots\ldots\ldots(4)$$

3.6. Dataset

In this study, the “Stroke Prediction Dataset” obtained from the Kaggle repository was used. This dataset includes features such as age, gender, and smoking status of individuals [54]. The dataset contains a total of 11 features, consisting of 10 input features and 1 output feature, aimed at classifying and predicting stroke risk. The features and their descriptions are listed below:

Age: The age of the individual, represented as numerical data.

Gender: Specifies the gender of the individual, represented as categorical data.

Hypertension: Indicates whether the individual has hypertension; represented as numerical data (1 – Hypertension present, 0 – No hypertension).

Work Type: Specifies the occupation of the individual, represented as categorical data (e.g., Government, Not working, Self-employed).

Heart Disease: Indicates whether the individual has heart disease; represented as numerical data (0 – No heart disease, 1 – heart disease present).

Marital Status: Shows the marital status of the individual, represented as categorical data (Yes, No).

Residence Area: Specifies the area where the individual lives, represented as categorical data (Urban, Rural).

Body Mass Index (BMI): Represents the body mass index of the individual, represented as numerical data.

Average Glucose Level: Indicates the average glucose level in the individual's blood, represented as numerical data.

Smoking Status: Specifies the smoking status of the individual, represented as categorical data (Previously smoked, currently smoking, never smoked).

Stroke Status: Indicates whether the individual has previously had a stroke; represented as numerical data (1 – Had a stroke, 0 – Did not have a stroke).

This dataset has been used in various academic research and theses, providing valuable information for assessing stroke risk [54].

4. Findings and Discussion

In the study, ML methods such as LR, KNN, SVM, and DT were used to detect the presence of stroke risk. To facilitate the readability of the results, performance metrics were visualized using the “Model Evaluation Matrix” and the “Confusion Matrix”.

In the LR model, a moderate regularization with C = 0.6 has been applied, which helps prevent overfitting. L1 penalty (Lasso) has been chosen using penalty = '11', which simplifies the model by reducing some features to zero. Additionally, solver = 'liblinear' has been selected, which is a suitable and effective solver, especially for small datasets.

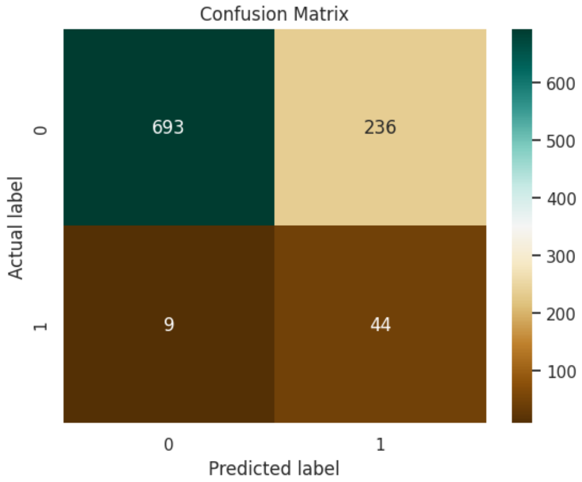

Figure 1. Confusion Matrix of the LR Model

Examining Figure 1, it can be seen that True Negatives (TN) (693) are correctly predicted. This high number of true negatives indicates that the model performs well in predicting negative classes. False Positives (FP) (236) are instances where true negatives were incorrectly predicted as positive. This indicates that the model misclassified some negative examples as positive. False Negatives (FN) (9) are instances where true positives were incorrectly predicted as negative. This suggests that the model missed some positive classes by predicting them as negative. True Positives (TP) (44) are instances where positive classes were correctly predicted. This shows that the model is successful in identifying positive classes, though there is room for improvement.

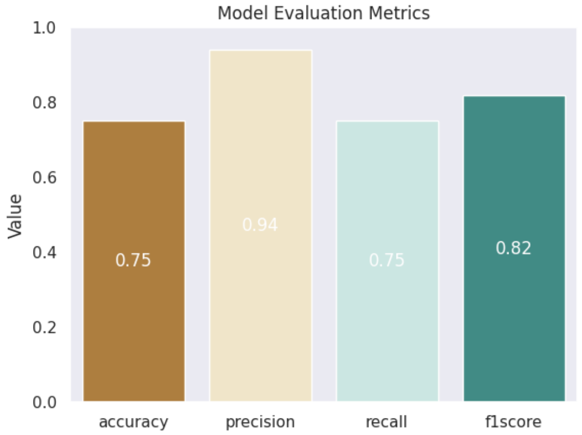

Figure 2. Model Evaluation Matrix of the LR Model

Examining Figure 2, it is observed that your model achieves high precision in classifying positive classes but may miss some positive examples, resulting in a moderate overall accuracy.

In the KNN model, the 'manhattan' metric has been identified as the most suitable option for measuring the distance between neighbors. With n_neighbors = 3, the best results were achieved by considering 3 neighbors. Furthermore, weights = 'distance' has been used to weight the neighbors based on their distances, meaning that closer neighbors have a greater impact on the prediction. With these hyperparameters, the model has achieved optimal performance based on the distance of neighbors.

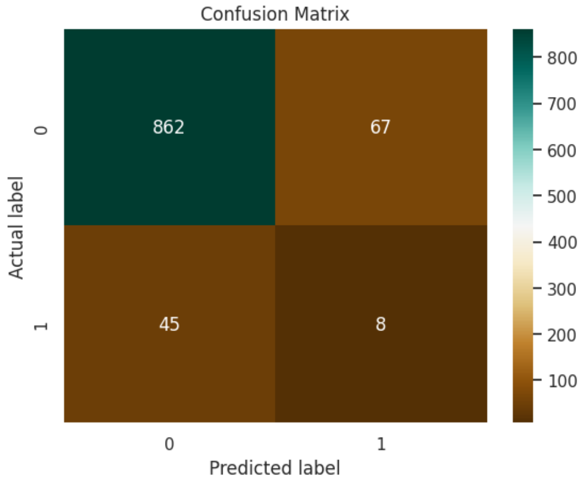

Figure 3. Confusion Matrix of the KNN Model

Figure 3 shows that TN (862) correctly predicts the true negative classes. This high true negative value indicates that the model predicts negative classes well. FP (67) shows that some true negatives are predicted as positive. This indicates that the model makes errors in predicting negative classes as positive. FN (45) shows that some true positives are predicted as negative. This means the model mistakenly predicts positive classes as negative, thus missing positive classes. TP (8) shows that the model correctly predicts true positives. This low true positive value indicates that the model is weak in identifying positive classes.

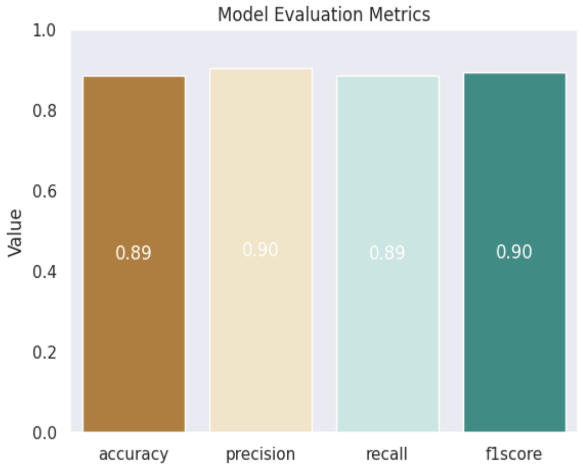

Figure 4. Model Evaluation Matrix of the KNN Model

Figure 4 shows that the model performs well overall, providing high precision and recall for positive classes and maintaining a good balance between precision and recall.

In the SVM model, the hyperparameters include a penalty parameter C = 10, which allows the model to learn a more complex boundary with a higher penalty. The 'rbf' kernel function has been chosen as it is effective on non-linear data. Additionally, the class_weight = 'balanced' setting attempts to balance the classes, optimizing the model’s performance according to the proportion of each class in the dataset.

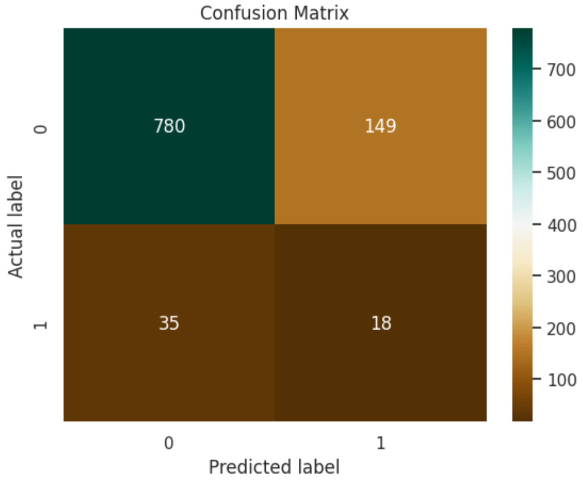

Figure 5. Confusion Matrix of the SVM Model

Figure 5 shows that TN (780) correctly predicts the true negative classes. This high true negative value indicates that the model predicts negative classes well. FP (149) shows that some true negatives are predicted as positive. This indicates that the model makes errors in predicting negative examples as positive. FN (35) shows that some true positives are predicted as negative. This means the model mistakenly predicts some positive examples as negative, thus missing positive classes. TP (18) shows that the model correctly predicts true positives. This low true positive value indicates that the model is weak in identifying positive classes.

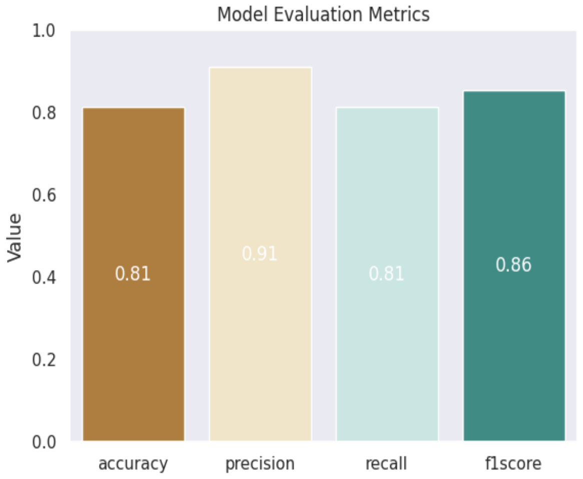

Figure 6. Model Evaluation Matrix of the SVM Model

Figure 6 shows that the model generally performs well, with high precision and strength in identifying positive classes. However, with a sensitivity of 81%, it suggests that the model might miss some positive examples.

For the KA model, the 'gini' impurity is used as the splitting criterion, which aims to increase the homogeneity of class distributions during splitting. The tree can grow up to a maximum depth of 20 levels (max_depth = 20), meaning it can have up to 20 branching layers. Additionally, at least 1 sample can be present in a leaf node (min_samples_leaf = 1).

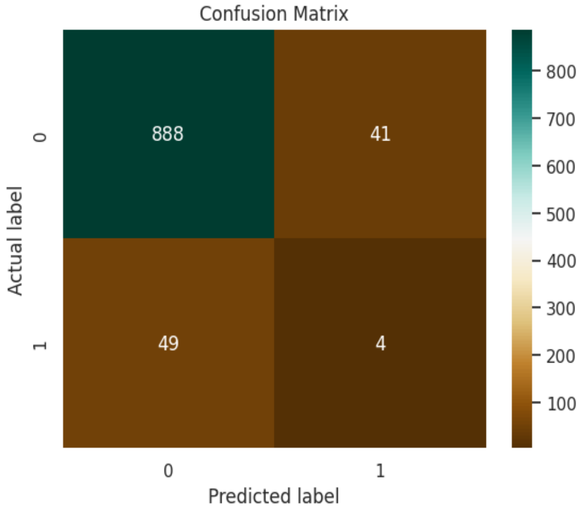

Figure 7. Confusion Matrix of the DT Model

Figure 7 shows that TN (888) correctly predicts the true negative classes. This high true negative value indicates that the model predicts negative classes very well. FP (41) shows that some true negatives are predicted as positive. This indicates that the model mistakenly predicts some negative examples as positive. FN (49) shows that some true positives are predicted as negative. This means the model mistakenly predicts some positive examples as negative, thus missing positive classes. TP (4) shows that the model correctly predicts true positives. This low true positive value indicates that the model is quite weak in identifying positive classes.

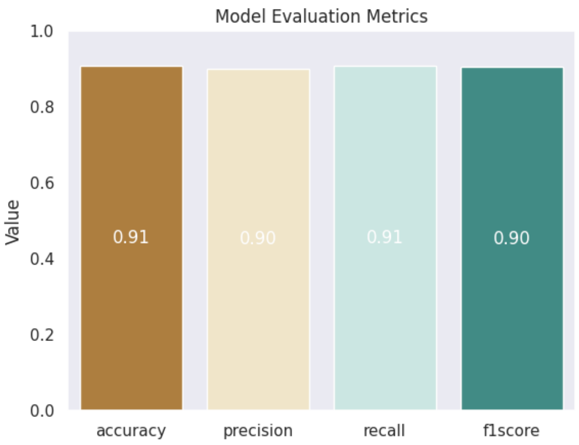

Figure 8. Model Evaluation Matrix of the DT Model

Table 2. Comparison of Performance Metrics for Used ML Models

Model	Class	Precision	Recall	F1-Score	Accuracy (%)
LR	No Risk	0.98	0.74	0.84	75,23
	Risk Present	0.15	0.83	0.26	
KNN	No Risk	0.95	0.92	0.92	88,83
	Risk Present	0.10	0.15	0.15	
SVM	No Risk	0.95	0.83	0.89	81,02
	Risk Present	0.10	0.33	0.16	
DT	No Risk	0.94	0.95	0.95	91,02
	Risk Present	0.08	0.07	0.08	

Figure 8 shows that the model has both high overall performance and high success in identifying positive classes. The model's results are quite satisfactory and can generally be considered reliable.

Table 2 compares the performance of the ML models used in the study for the “Risk Present” and “No Risk” classes. The LR model shows high precision in recognizing the “No Risk” class. However, recall is somewhat lower, indicating that some “Risk Present” cases may be missed. The F1-score and overall accuracy reflect this situation. The KNN algorithm demonstrates high performance for the “No Risk” class. Both precision and recall are quite high, resulting in good overall accuracy for the model. However, it shows poor performance in the “Risk Present” class, with both precision and recall being low, indicating difficulty in distinguishing this class. The SVM model exhibits very good performance for the “No Risk” class, with high F1-score and accuracy, though recall is somewhat lower. It is noted for having the highest performance in the “No Risk” class, with both precision and recall being quite high and the F1-score being close to perfection.

5. Conclusion and Suggestions

This study investigated the risk factors associated with stroke and assessed their impact on the development of the disease. The findings indicate that hypertension, diabetes, smoking, and high cholesterol levels increase stroke risk, with hypertension being the most significant determinant. Additionally, it was found that ischemic stroke is more prevalent, whereas hemorrhagic stroke is associated with higher mortality rates. The critical role of early diagnosis and intervention, particularly thrombolytic therapy and rehabilitation, in the recovery process of patients was emphasized.

The study developed a method for stroke detection and diagnosis using ML models and demonstrated that these models contribute to medical decision support systems. The model's accuracy was reported to be 91%. Future studies aim to enhance success by employing different ML models and performance metrics, with more effective feature selection and data preprocessing steps.

It is recommended to explore additional ML models and advanced techniques to improve the accuracy and reliability of stroke risk prediction. Experimenting with ensemble methods and deep learning algorithms could provide better insights and enhance model performance.

Future research should focus on identifying and incorporating additional relevant features that could influence stroke risk. This may involve integrating new biomarkers, genetic factors, or lifestyle variables to create more comprehensive models.

Emphasizing rigorous data preprocessing and cleaning is crucial to ensure the quality and consistency of the data used in model training. Addressing issues such as missing values, outliers, and data imbalance will help improve model robustness.

It is important to validate the models on diverse and independent datasets to ensure their generalizability and applicability in different populations. Cross-validation techniques and external validation studies should be employed to confirm the model's effectiveness.

Efforts should be made to integrate these predictive models into clinical decision support systems to assist healthcare professionals in early stroke detection and management. Providing user-friendly interfaces and decision aids will enhance the practical utility of these models.

Regularly updating the models with new data and research findings is essential to maintain their relevance and accuracy. Continuous monitoring and iterative improvements based on real-world performance will contribute to better patient outcomes.

By addressing these recommendations, the accuracy and impact of stroke risk prediction models can be further improved, leading to more effective prevention and management strategies.

Contributions of the authors

A.Ç. and H.A. designed the study, performed the experiments and wrote the article. H.A. performed the calculations, A.Ç checked the language and contributed to the writing of the manuscript.

Conflict of Interest Statement

There is no conflict of interest between the authors.

Statement of Research and Publication Ethics

The study is complied with research and publication ethics.

References

[1] S. Emon, M. M. Talukder, M. M. Rahman, and M. S. Islam, "Stroke prediction using machine learning algorithms," in *2020 IEEE Region 10 Symposium (TENSYMP)*, Dhaka, Bangladesh, 2020, pp. 101–104.

[2] M. Singh, V. S. Chouhan, and S. Gupta, "Prediction of stroke using artificial intelligence," in *2017 International Conference on Computing, Communication and Automation (ICCCA)*, Greater Noida, India, 2017, pp. 1467–1471.

[3] E. Sevli, "Prediction of stroke using Random Forest Classifier," in *2020 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, Sakheer, Bahrain, 2020, pp. 1–6.

[4] K. Revanth, G. Deepak, and K. N. V. Satya Prasad, "A comparative study on stroke prediction using various machine learning models," in *2020 International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, 2020, pp. 1385–1390.

[5] S. Cheon, D. A. Lee, and J. K. Kim, "Detection of stroke using deep learning algorithms," in *2019 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, USA, 2019, pp. 1–4.

[6] S. Shoily, A. B. K. Singh, and M. Rahman, “Analysis of machine learning algorithms for stroke detection,” in *2019 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)*, Rajshahi, Bangladesh, 2019, pp. 1–4.

[7] N. Pradeepa, R. Gupta, and R. Singh, “Detection of stroke symptoms using machine learning and social media analytics,” in *2020 IEEE Symposium on Signal Processing and Information Technology (ISSPIT)*, Bilbao, Spain, 2020, pp. 1–6.

[8] H. Li, J. Liu, and Y. Wang, “Stroke risk prediction with various machine learning algorithms,” in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, San Diego, CA, USA, 2019, pp. 2148–2151.

[9] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[10] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2012, pp. 1106–1114.

[11] L. Zhang, "Machine Learning Techniques for Big Data Analysis," Ph.D. dissertation, Dept. of Computer Science, University of California, Berkeley, CA, USA, 2018.

[12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, Cambridge, MA: MIT Press, 2016.

[13] M. U. Emon, M. S. Keya, T. I. Meghla, M. M. Rahman, S. Al Mamun, and M. S. Kaiser, “Performance analysis of machine learning,” in *Fourth International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, 2020, pp. 1464–1469.

[14] M. S. Singh and P. Choudhary, “Stroke prediction using artificial intelligence,” in *8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON)*, Bangkok, Thailand, 2017, pp. 158–161.

[15] O. Sevli, “İnme (felç) riskinin MÖ kullanılarak tespiti,” in *7. Uluslararası Mühendislik Mimarlık ve Tasarım Kongresi (7th International Congress on Engineering, Architecture and Design)*, İstanbul, Türkiye, 2021, pp. 661–667.

[16] S. Revanth, S. Sanjay, and V. Vijagayaganth, “Stroke prediction using machine learning algorithms,” *International Journal of Disaster Recovery and Business Continuity*, vol. 11, no. 1, pp. 3081–3086, 2020.

[17] S. Cheon, J. Kim, and J. Lim, “The use of deep learning to predict stroke patient mortality,” *International Journal of Environmental Research and Public Health*, vol. 16, no. 11, pp. 1–12, 2019.

[18] T. I. Shoily, T. Islam, S. Jannat, S. A. Tanna, T. M. Alif, and R. R. Ema, “Detection of stroke disease using machine learning algorithms,” in *Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, IEEE, 2019, pp. 1–6.

[19] S. Pradeepa, K. Manjula, S. Vimal, M. S. Khan, N. Chilamkurti, and A. K. Luhach, “DRFS: Detecting risk factor of stroke disease from social media using machine learning techniques,” *Neural Processing Letters*, 2020, pp. 1–19.

[20] X. Li, D. Bian, J. Yu, M. Li, and D. Zhao, “Using machine learning models to improve stroke risk level classification methods of China national stroke screening,” *BMC Medical Informatics and Decision Making*, vol. 19, pp. 1–7, 2019.

[21] D. R. Cox, “Regression models and life-tables,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 34, no. 2, pp. 187–220, 1972.

[22] M. E. T. M. MacKinnon, “Logistic regression analysis for binary data,” *Journal of Applied Statistics*, vol. 28, no. 6, pp. 759–769, Jun. 2001.

[23] A. Agresti, “Categorical data analysis,” *Journal of the American Statistical Association*, vol. 95, no. 450, pp. 1121–1130, Dec. 2000.

[24] S. M. Bhandari, J. P. Lee, and K. J. Kim, “Application of logistic regression in predicting heart disease,” in *Proc. IEEE International Conference on Data Mining (ICDM)*, Dec. 2019, pp. 1124–1130.

[25] S. Wang and X. Zhang, “Improving logistic regression with feature selection for high-dimensional data,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018, pp. 1552–1560.

[26] L. Huang and X. Liu, “Logistic regression for multiclass classification problems,” in *Proc. IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2017, pp. 987–994.

[27] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[28] M. A. Hall and L. A. Smith, “Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper,” *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 103–110, Sep. 2003.

[29] D. Zhang, M. M. Islam, and G. Lu, “A review on automatic image annotation techniques,” *Pattern Recognition*, vol. 45, no. 1, pp. 346–362, Jan. 2012.

[30] K. Q. Weinberger and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification,” in *Proc. Advances in Neural Information Processing Systems (NIPS)*, Dec. 2006, pp. 1473–1480.

[31] A. Beygelzimer, S. Kakade, and J. Langford, “Cover trees for nearest neighbor,” in *Proc. 23rd International Conference on Machine Learning (ICML)*, Jun. 2006, pp. 97–104.

[32] M. Muja and D. G. Lowe, “Fast approximate nearest neighbors with automatic algorithm configuration,” in *Proc. IEEE International Conference on Computer Vision Theory and Applications (VISAPP)*, Feb. 2009, pp. 331–340.

[33] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[34] J. Weston and C. Watkins, “Support vector machines for multi-class pattern recognition,” in *European Symposium on Artificial Neural Networks (ESANN)*, vol. 4, pp. 219–224, Apr. 1999.

[35] B. Scholkopf, A. Smola, R. Williamson, and P. Bartlett, “New support vector algorithms,” *Neural Computation*, vol. 12, no. 5, pp. 1207–1245, May 2000.

[36] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge: Cambridge University Press, 2000.

[37] B. Boser, I. Guyon, and V. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proc. 5th Annual ACM Workshop on Computational Learning Theory (COLT)*, Jul. 1992, pp. 144–152.

[38] A. J. Smola and B. Scholkopf, “A tutorial on support vector regression,” in *Proc. International Conference on Artificial Neural Networks (ICANN)*, Aug. 1998, pp. 52–78.

[39] J. R. Quinlan, “Induction of decision trees,” *Machine Learning*, vol. 1, no. 1, pp. 81–106, Mar. 1986.

[40] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[41] S. Murthy, “Automatic construction of decision trees from data: A multi-disciplinary survey,” *Data Mining and Knowledge Discovery*, vol. 2, no. 4, pp. 345–389, Dec. 1998.

[42] R. Kohavi, “Scaling up the accuracy of Naive-Bayes classifiers: A decision-tree hybrid,” in *Proc. 2nd International Conference on Knowledge Discovery and Data Mining (KDD)*, Aug. 1996, pp. 202–207.

[43] P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees,” in *Proc. European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, Sep. 2006, pp. 3–42.

[44] H. Blockeel and L. De Raedt, “Top-down induction of first-order logical decision trees,” in *Proc. 9th International Conference on Machine Learning (ICML)*, Jul. 1998, pp. 55–63.

[45] V. Tümen and A. S. Sunar, “Predicting the work-life balance of employees based on the ensemble learning method,” *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, vol. 12, no. 2, pp. 344–353, 2023.

[46] A. Çalışkan, “Finding complement of inefficient feature clusters obtained by metaheuristic optimization algorithms to detect rock mineral types,” *Transactions of the Institute of Measurement and Control*, vol. 45, no. 10, pp. 1815–1828, 2023.

[47] D. M. W. Powers, “Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation,” *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[48] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, Jul. 2009.

[49] S. B. Çelebi and Ö. A. Karaman, “Multilayer LSTM model for wind power estimation in the SCADA system,” *European Journal of Technique (EJT)*, vol. 13, no. 2, pp. 116–122, 2023.

[50] J. Davis and M. Goadrich, “The relationship between precision-recall and ROC curves,” in *Proc. 23rd International Conference on Machine Learning (ICML)*, Pittsburgh, PA, USA, Jun. 2006, pp. 233–240.

[51] V. Tümen, “SpiCoNET: A hybrid deep learning model to diagnose COVID-19 and pneumonia using chest X-ray images,” *Traitement du Signal*, vol. 39, no. 4, pp. 1169, 2022.

[52] N. Chinchor, “MUC-4 evaluation metrics,” in *Proc. 4th Conference on Message Understanding*, McLean, VA, USA, Jun. 1992, pp. 22–29.

[53] A. Çalışkan, “Classification of tympanic membrane images based on VGG16 model,” *Kocaeli Journal of Science and Engineering*, vol. 5, no. 1, pp. 105–111, 2022.

[54] Kaggle, "Stroke Prediction Dataset," [Online]. Available: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset. [Accessed: May 8, 2024].