Analyzing the Performance of Brain Stroke Prediction Using Various Machine Learning Classification Algorithms

Zuhayr Mahrus Kabir

Abstract

In today's day and age, strokes are officially regarded as the leading cause of death and disability globally, per the World Health Organization (WHO). A stroke, also known as a cerebrovascular accident, is a medical emergency that occurs when blood flow to a part of the brain is interrupted or reduced, leading to damage or death of brain cells [1]. In many cases, the medical diagnosis of a stroke isn't attained until after its onset, which, more often than not, leads to fatal consequences. Prompt medical attention is critical when treating a stroke to minimize brain damage and prevent long-term disability or death. Lately, machine learning has been viewed as a significant advancement towards preemptive stroke diagnosis [2]. Machine learning (ML) algorithms analyze vast amounts of medical data, including electronic health records, medical imaging, genetic information, and real-time patient monitoring data, to uncover patterns and insights that were previously unattainable [3]. This research paper investigates the application of machine learning models at their fundamental level for stroke prediction. The paper employs a supervised machine learning model, applying regression algorithms to a collected patient dataset comprising demographic, clinical, and lifestyle factors of patients. Various classifiers, including logistic regression, decision trees, support vector machines (SVM), k-nearest neighbors (KNN), and random forest, were employed to develop predictive models. The study aimed to assess the performance of these classifiers and identify the most accurate model for stroke prediction. Results indicated that the random forest classifier achieved the highest accuracy among all models evaluated, with 99.81% accuracy. This finding underscores the efficacy of ensemble learning techniques in capturing complex interactions and non-linear relationships within the data. The research highlights the potential of ML-based approaches for identifying high-risk individuals for stroke and guiding targeted preventive interventions in clinical practice.

Introduction

Stroke, a cerebrovascular accident, is the leading cause of death and disability. A stroke occurs when the blood supply to part of the brain is interrupted or reduced, preventing brain tissue from receiving oxygen and nutrients, which can lead to the death of brain cells within minutes. Strokes can be classified as either ischemic, caused by blockages or narrowing of the arteries supplying blood to the brain, or hemorrhagic, resulting from blood vessels in the brain bursting and causing bleeding [4]. The severity and outcomes of a stroke can vary widely, ranging from complete recovery to long-term disability or death, depending on the location and extent of brain tissue affected.

The World Health Organization states, “15 million people worldwide suffer a stroke. Of these, 5 million die and another 5 million are left permanently disabled, placing a burden on family and community.” [5] Strokes may affect people of all ages, although the likelihood of stroke onset skyrockets for people aged 55 and above ("Stroke Facts & Statistics"). Furthermore, the WHO states that from 1990 to 2019, there has been a 70% increase in stroke incidence, of which a 43% increase in fatality rate has transpired. In the United States itself, it is estimated that close to 800,000 people are affected by a stroke annually as of 2024.

The onset of this disease is influenced by a myriad of lifestyle factors and pre-existing conditions. Among the most critical risk factors are high blood pressure, diabetes, and heart diseases such as atrial fibrillation. Cigarette smoking is also a major contributor. Additional risk factors encompass physical inactivity, being overweight or obese, and having high cholesterol levels. Conditions like sickle cell disease and excessive alcohol consumption further elevate stroke risk. A family history of stroke, drug abuse, and genetic conditions such as blood-clotting disorders or vascular disorders are also notable contributors [6]. Understanding these diverse risk factors is crucial for developing comprehensive predictive models and effective preventive strategies.

To combat the repercussions of stroke onsets, the ability to accurately predict stroke diagnosis plays a crucial role in preventive medicine, thus enabling early intervention strategies to mitigate its devastating consequences.

Unfortunately, existing methods for stroke prediction often face challenges in achieving high accuracy and reliability, leading to missed opportunities for timely intervention. Recently, however, machine learning has been widely implemented in the medical field. By accumulating patient data, ML algorithms can discover patterns in datasets that allow medical professionals to predict treatment outcomes. Multiple studies have already been conducted regarding the potential of utilizing machine learning to forecast stroke; yet, it has not been widely established as a medical measure as of yet. The potential of ML within healthcare nonetheless continues to intrigue, and this paper seeks to reinforce this idea.

As a lot of research and investment has already been dedicated to heart stroke predictions, this paper aims to diverge and address machine learning techniques for predicting the onset of brain stroke. The underlying problem in the domain of stroke prediction lies in the limitations of traditional risk assessment methods, which rely on manual scoring systems or simplistic statistical approaches. These methods often fail to capture the complex interplay of risk factors and may overlook subtle patterns indicative of impending stroke events. As a result, there is a pressing need for more sophisticated and accurate predictive models that can identify individuals at heightened risk of stroke with greater precision and reliability. However, for this

exploration, the data utilized is very limited in that it does not factor in images for classifying whether a patient has the chance for the onset of a stroke or not, but rather depends on demographic, lifestyle, and biological data.

The significance of this research lies in its potential to revolutionize stroke risk assessment and preventive healthcare practices. By harnessing the power of ML algorithms, software engineers may aim to develop predictive models capable of analyzing vast datasets of patient information to identify subtle indicators of stroke risk [7], [8], [9]. This research has profound implications for clinical practice, as accurate stroke prediction can enable healthcare providers to implement timely interventions, such as lifestyle modifications, medication management, and targeted medical interventions, to prevent or minimize the occurrence of strokes and associated disabilities.

Methodology

Proposed Process of Exploration

To conduct this investigation, the dataset must first be processed. The dataset is available for model construction following its processing, through which a variety of prominent ML models will be trained to "learn" the data. After creating and training the models, several accuracy measures will be utilized to compare the success of the trained models. Below, a general overview of the Machine Learning workflow is documented and will be followed in this exploration.

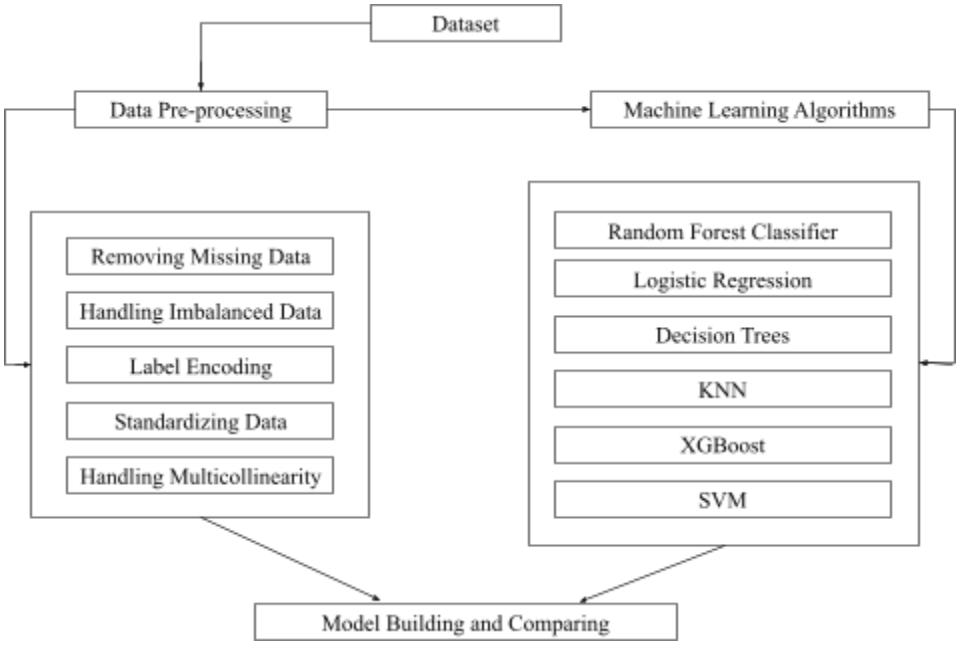

Figure 1: Machine Learning Workflow

Data Set

The data set is a .csv file, containing clinical features from 40910 observations with 11 different attributes. It has been collected from a Kaggle dataset, with both numerical and categorical values. The 11 attributes of the data can be seen in Table 1 below:

No.	Name of Attribute	Description of Attribute	Type of Data (Numerical/Categorical)
1	sex	patient's gender (1: male; 0: female)	Categorical
2	age	patient's age (in years)	Numerical
3	hypertension	patient has ever had hypertension (1) or not (0)	Categorical
4	heart_disease	patient has ever had heart_disease(1) or not (0)	Categorical
5	ever_married	patient married (1) or not (0)	Categorical
6	work_type	patient job type: 0 - Never_worked, 1 - children, 2 - Govt_job, 3 - Self-employed, 4 - Private	Categorical
7	Residence_type	patient area: 1 - Urban, 0 - Rural	Categorical
8	avg_glucose_level	patient average blood sugar level	Numerical
9	bmi	Body Mass Index	Numerical
10	smoking_status	1 - smokes, 0 - never smoked	Categorical
11	stroke	Whether the patient has stroke (1) or not (0)	Categorical

Table 1: Dataset + Descriptions

The first 10 rows constitute the input hypermaters that will be inputted into various machine learning models, namely the patient's: sex (categorical), age (numberical), hypertension history (categorical), heart disease history (categorical), marriage history (categorical), work status (categorical), residence type (categorical), average blood glucose levels (numerical), current body-mass-index (numerical), and smoking status (categorical).

The 11th row above signifies the response variable, which the developed machine learning models aim to predict, whereas the remaining rows indicate the exploratory variables for this investigation.

Data Pre-processing

Data pre-processing is a necessary component of the machine learning workflow, utilized to convert raw data into a more useful and filtered format to improve data quality by removing any unwanted noise and outliers that could deviate the model from its intended training.

The primary step taken is to determine whether any of the parameters may be deemed to be irrelevant for stroke prediction. As ML models are best trained when they are tuned with the necessary parameters, it is essential to remove unimportant features. Additionally, any null response variable data points (stroke or no stroke) must be accordingly removed. Patient data containing null parameter values should be filled up by mean values as an estimate.

Thereafter, the minority class of the response variable (if a minority exists) must be oversampled using the Synthetic Minority Over-sampling Technique (SMOTE) if there exists a drastic imbalance in the occurrence of the target variable (number of patients diagnosed with stroke) as opposed to the absence of the target variable or vice-versa. Having a balanced data set is indispensable to preventing bias in the model.

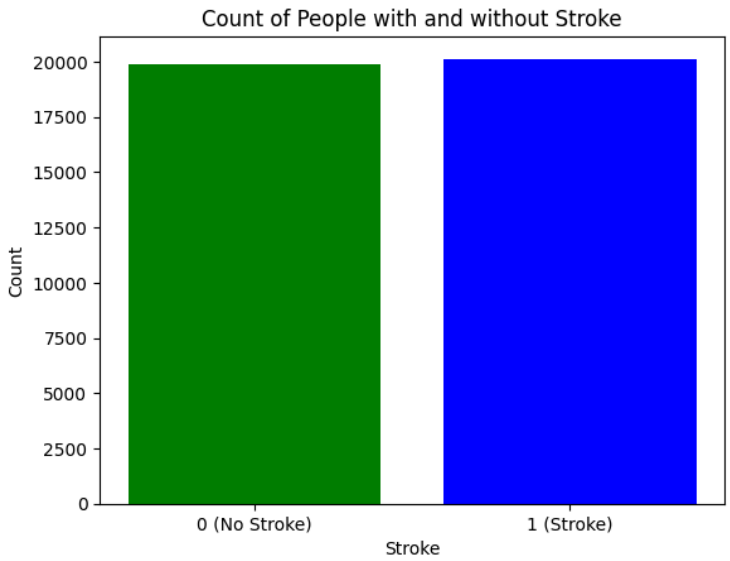

Figure 1: Stroke Patient Count

Based on Figure 1:

# of Patients WITH Stroke	Number of Patients WITHOUT Stroke
20128	19857

The data set for the target variable (stroke) is quite balanced: 50.34% of the patients from the data set have had the onset of stroke, whereas the remaining 49.66% have not been diagnosed with stroke. As a result, SMOTE is not required to address any class imbalance problems.

Typically, label encoding must first be performed to ensure that any string-literal categorical values are converted into a numerical format; however, this data set has been published in a feature-encoded format by default, namely having been converted into a numerical format through the One Hot Encoding technique.

Additionally, the data must be modified such that the parameters are standardized through appropriate scaling, allowing all numerical values to be comparable on a similar axis. Standardization occurs by transforming the features to have a mean close to 0 and a standard deviation close to 1. This technique transforms the values of features to a similar scale, ensuring that feature contributions to model predictions are equally significant.

The data is then run through a Z-score test to identify if any outliers exist. A Z-score indicates the number of standard deviations a value is from the mean of a data set's distribution; any point with a z-score having a magnitude greater than or equal to 3 indicates an outlier point. These outliers are then removed.

The final key measure taken for data pre-processing is to ensure that the data are not too correlated, as multicollinearity can lead to unstable and unreliable coefficient estimates in regression models and perhaps model overfitting to a more extreme extent, reducing the model's interoperability.

Machine Learning Classifiers

In this paper, 6 different machine classifiers were used: Logistic Regression, Decision Tree, Random Forest Classifier, SVM, KNN, and XGBoost. The data has to be broken up into predominantly a training set and the rest into a testing set. Validating the ML models is done using the k-fold cross-validation technique. This evaluates the performance of an ML model and ensures its ability to generalize to new, unseen data. The dataset is partitioned into k subsets, or "folds". The models are trained k times, where a different fold is used as the validation set each time, and the remaining k-1 folds are used as the training set. This process helps mitigate issues related to overfitting. The performance metrics, namely the accuracy, AUC, and

F1-scores, are calculated for each iteration, and the results are averaged to provide a more reliable estimate of the model's true performance. K-fold cross-validation ensures that every data point is used for both training and validation, making it a robust method for performance evaluation and hyperparameter tuning in machine learning models.

Exploratory Data Analysis

Data Visualization

	sex	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status
count	40907.000000	40907.000000	40907.000000	40907.000000	40907.000000	40907.000000	40907.000000	40907.000000	40907.000000	40907.000000
mean	0.555162	51.327303	0.213851	0.127729	0.821326	3.461095	0.514851	122.079679	30.406488	0.488572
std	0.496954	21.624171	0.410028	0.333792	0.383083	0.780934	0.499786	57.561951	6.835305	0.499875
min	0.000000	-9.000000	0.000000	0.000000	0.000000	0.000000	0.000000	55.120000	11.500000	0.000000
25%	0.000000	35.000000	0.000000	0.000000	1.000000	3.000000	0.000000	78.750000	25.900000	0.000000
50%	1.000000	52.000000	0.000000	0.000000	1.000000	4.000000	1.000000	97.920000	29.400000	0.000000
75%	1.000000	68.000000	0.000000	0.000000	1.000000	4.000000	1.000000	167.590000	34.100000	1.000000
max	1.000000	103.000000	1.000000	1.000000	1.000000	4.000000	1.000000	271.740000	92.000000	1.000000

Figure 2: Initial Summary Statistics of the Data Set Features

Figure 2 presents the summary statistics of the different features of the data set, indicating data such as the mean value, the standard deviation, the minimum and maximum values, as well as the quartile values of the feature.

Histograms are an effective way to depict the frequency of the different features of the data set. Figure 3 illustrates the dataset's proportions.

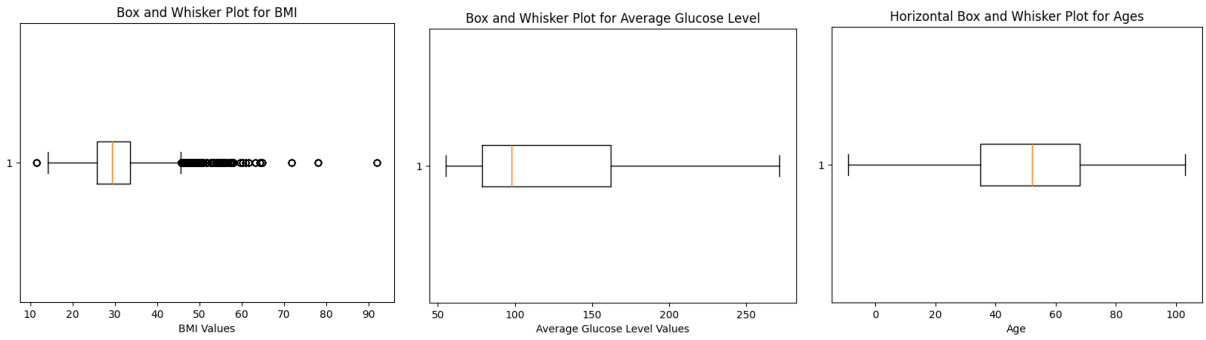

Figure 3: Box-and-Whisker Plots for BMI, Average Glucose Level, and Age

Box-and-whisker plots effectively convey key information about the data set, identifying the point of central tendency, as well as the distribution of data. As can be seen from the box-and-whisker plots in Figure 3, only the BMI values contain some significant outliers. Removing these outliers using the aforementioned z-score technique will be performed in the *Data Pre-processing* stage.

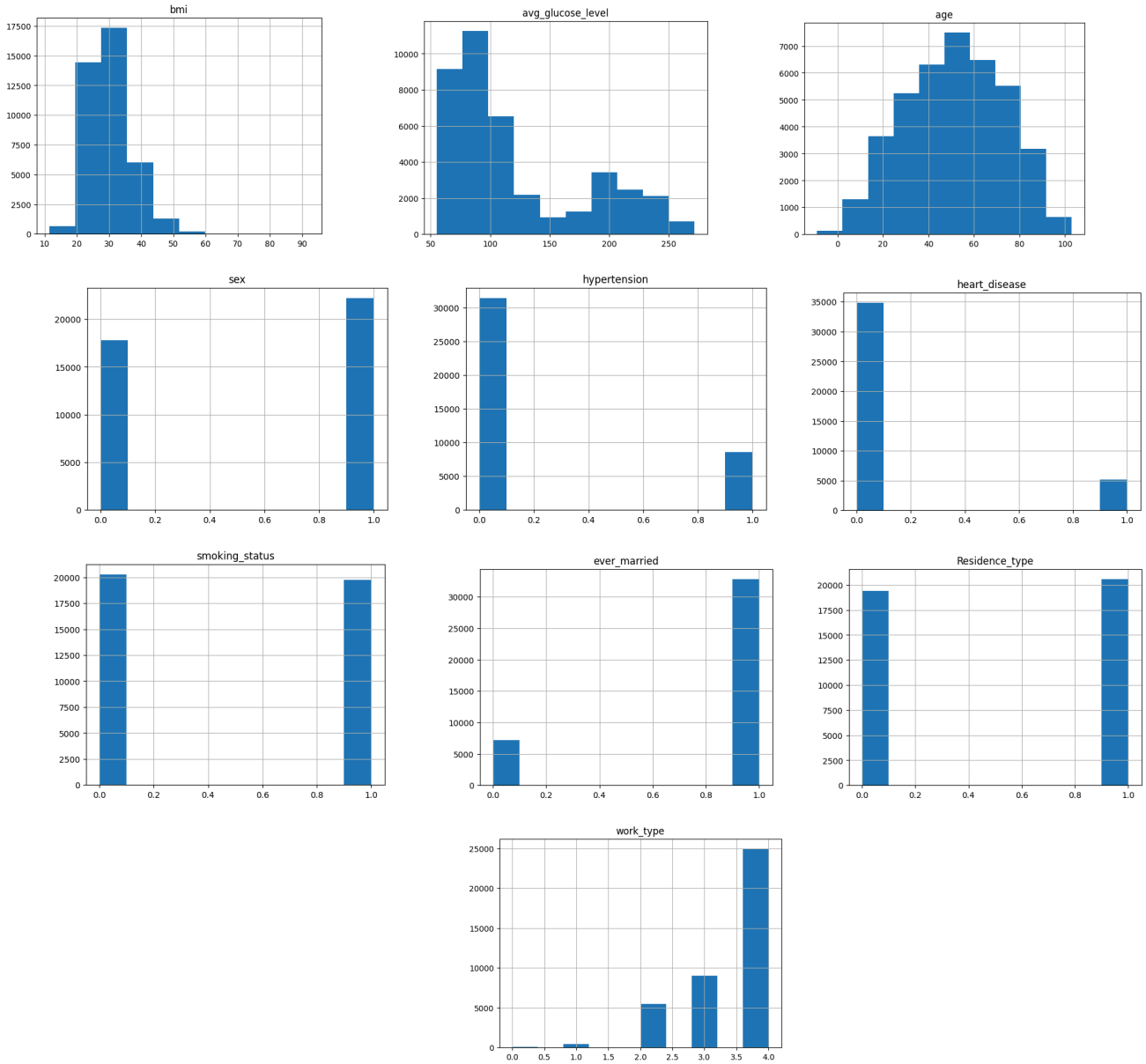

Figure 4: Histogram of Features in the Data Set

It is crucial to note from Figure 4 that the categories of hypertension, heart_disease, ever_married are all instances of imbalanced data, which may affect feature selection and cause biases in the trained models.

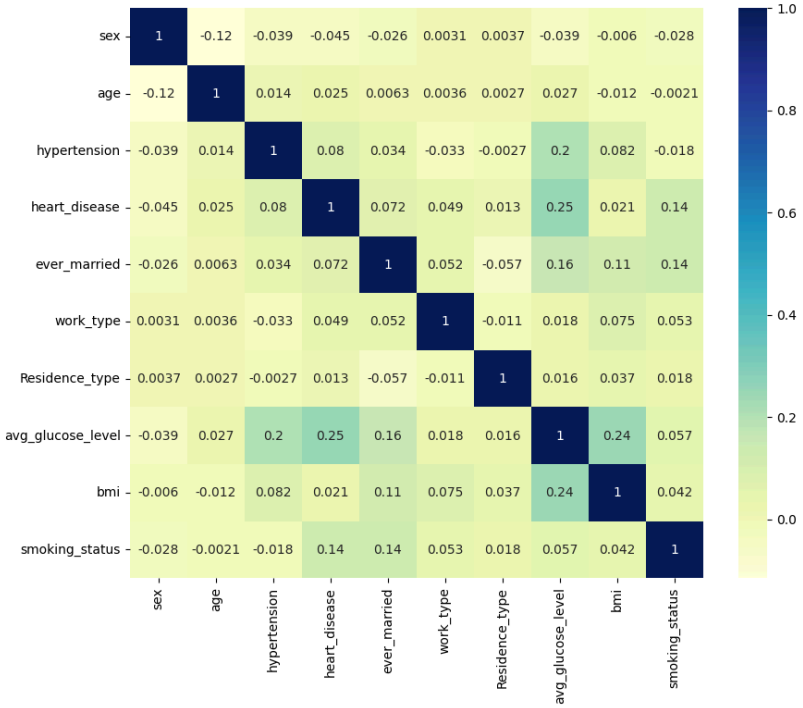

Figure 5: Correlation Heatmap between Different Parameters

The correlation between pairs of features depicted in Figure 5 indicates the strength and direction of the linear relationship between them. A correlation coefficient provides a statistical measure of how changes in one of the variables result in changes in the other. A highly correlated pair refers to two features or variables in a dataset that exhibit a strong linear relationship with each other, indicating a close connection and implying that as one variable changes, the other tends to change in a consistent and predictable manner.

A correlation coefficient typically ranges from -1 to 1:

- 1 indicates a perfect positive correlation (both variables increase or decrease together)
- -1 represents a perfect negative correlation (one variable increases as the other decreases)
- 0 implies no linear correlation.

Any pair with a correlation value past the threshold of ± 0.8 is conventionally accepted to be a highly correlated pair and should be removed from the dataset so as to minimize the effects of multicollinearity, as it may lead to skewed or misleading results. If two features are nearly identical or strongly correlated, they might carry redundant information, potentially leading to issues like multicollinearity. Multicollinearity can impact the stability of regression coefficients and make it challenging for the model to discern the individual contribution of each feature.

Figure 5 shows that there are no two parameters that are highly correlated to one another; thus, multicollinearity should not affect the results produced in this investigation.

Data Pre-processing

By first removing any rows containing NULL values for any of the input parameters or for the target variable, there was some information loss, albeit minimal in the context of the sample data size. The initial data set contained 40910 rows or patient details; however, upon removing missing data records, there remained 40907 records, a mere 3 patient details being excluded, or 0.007% of the total data set. The data set was then standardized, with an average value of 0 and a standard deviation of 1.

Removing Outliers

As mentioned prior, through an investigation of the 3 primary numerical features, BMI is the only one that contains significant outliers to the data set. Figure 6 depicts the result of removing any such outliers.

Figure 6: Modified Box-and-Whisker Plots for BMI, Average Glucose Level, and Age (without outliers)

Visualization of Feature Selection

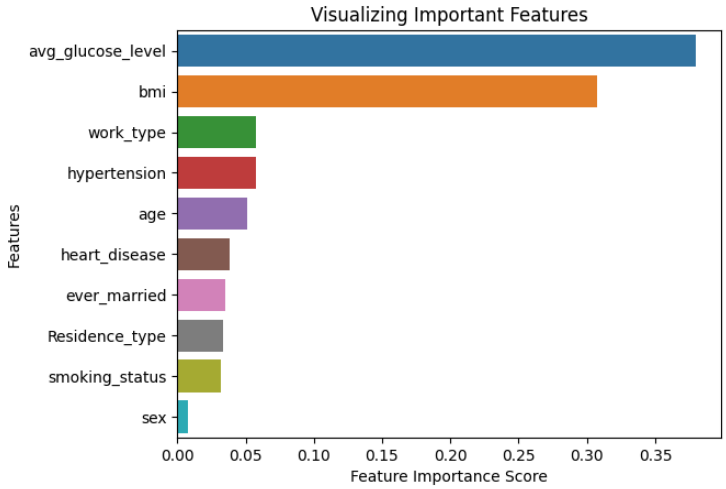

Figure 7: Feature Importance Score of the Parameters on Stroke

Figure 7 shows that all the parameters have a positive feature importance score. Feature importance is a technique whereby a “score” is calculated for each input feature of a model. The above image depicts the relationship between the input parameters and the testing variable based on the data set, with avg_glucose_level and bmi having the most profound influence on the onset of stroke. This score signifies the importance or weighting that is carried by that input parameter as the model returns predicted data based on the training data set input. A higher score indicates the input feature has a more profound effect on the model that is being used to predict a certain variable. It is crucial to note that feature importance calculates the significance of the features relative to one another as opposed to the importance of a single feature independently relative to the target variable. This step is crucial, as the ultimate goal is to identify and retain the most important features and discard the less important and redundant ones.

Data Splits

The next step is to divide the data set randomly into training and testing data. Multiple variations of the training testing split proportions were considered, such as 80-20% and 60-40% splits; however, for this investigation, the performance was optimized at a 70-30% case.

Evaluation/Confusion Matrix

A confusion matrix will be used to evaluate the performance of different machine learning classification algorithms. It illustrates the frequency with which the different models can accurately predict the outcome of the target variable [10]. True positive and true negative values are intended results, as they demonstrate accuracy in the models; however, false positive and false negative values, on the contrary, are indicative of inaccuracies in the prediction models. In this investigation, false positives may be more accepted than false negatives solely due to additional precautionary measures being taken regardless of a stroke diagnosis.

Through the use of the confusion matrices, the different models’ accuracy, precision-recall trade-off, F1 score, and AUC can be utilized to assess their testing performance. Formulas for calculation can be found in the appendices (A.1 through A.7).

Machine Learning Models

For this investigation, some of the most prominent algorithmic models, namely Decision Trees, Random Forest Classifier, XGBoost, Logistic Regression, KNN, and SVM were utilized to predict the diagnosis of stroke utilizing the aforesaid dataset.

Logistic Regression

A logistic regression model starts with a linear equation of the form:

$$z = b_n x^n + b_{n-1} x^{n-1} + \ldots + b_2 x^2 + b_1 x^1 + b_0$$

Here, z is the linear combination of features (indicated by $x^n, x^{n-1}, \ldots, x^2, x^1$) and is weighted by coefficients (b_n, b_{n-1}, …, b_2, b_1, b_0). [11]

The above linear function is converted into a probability using a sigmoid function, which is applied on z. This sigmoid function can be modelled by:

$$P(Y = 1) = \frac{1}{1+e^{-z}}$$

$P(Y = 1)$ is the probability of the positive class (patient being diagnosed with stroke). This logistic regression function serves to convert a linear model into a probability whereby a higher value (past the threshold value of 0.5) inclines the model towards predicting 1 (positive class), whereas a lower probability inclines the model's prediction towards 0 (negative class).

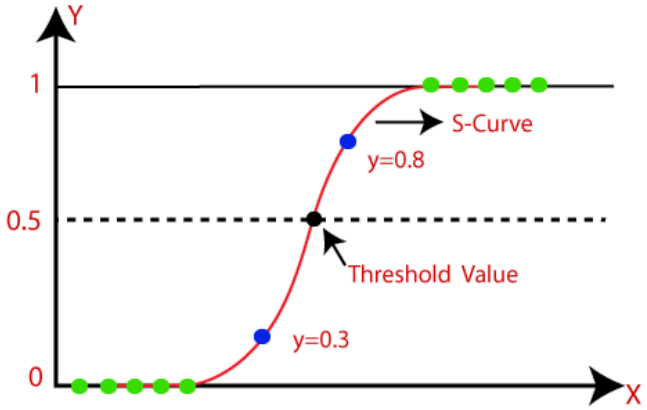

Figure 8: Logistic Regression Sigmoid Function (Song)

The logistic regression model learns the weights associated with each feature during training, adjusting them to maximize the likelihood of the observed outcomes. These learned weights, along with an intercept term, define the decision boundary, which typically tends to be 0.5.

Logistic regression is a simple, interpretable, and computationally efficient model. It performs well in binary classification tasks, providing probabilistic predictions that can be easily understood [12]. Logistic regression is particularly effective when the relationship between predictors and the target variable is roughly linear.

However, logistic regression may struggle when faced with non-linear patterns or complex interactions between features. Its simplicity can be a limitation in capturing intricate relationships present in the data.

Decision Trees

Decision Tree is an ML model that can be utilized for both regression and classification purposes. This model resembles a tree, operating by recursively partitioning the dataset into subsets based on the values of input features, ultimately making decisions or predictions based on the data's characteristics [13], [14].

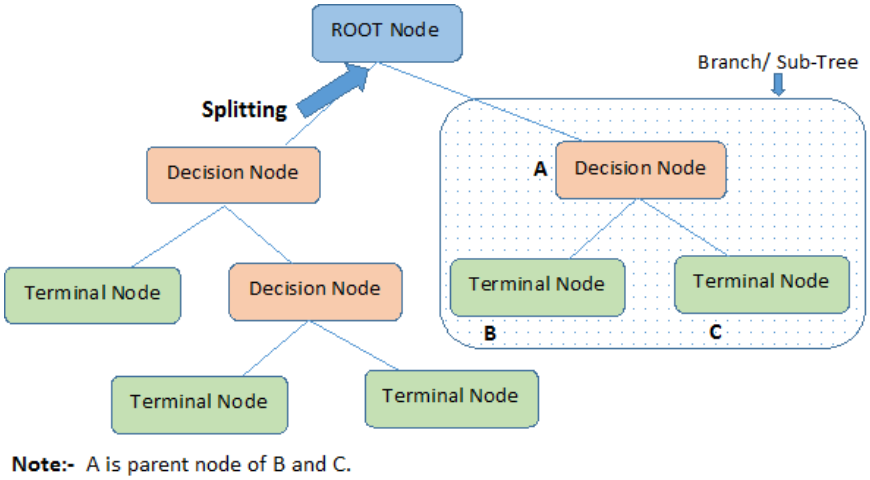

Figure 9: Decision Tree Workflow (Nicholson)

The process begins with the selection of a root node, representing the entire training dataset. Subsequent nodes in the tree correspond to feature splits, where the dataset is divided based on a chosen feature's values. The decision tree algorithm employs a top-down approach, selecting the feature that optimally splits the data at each node. This optimization is typically based on criteria like Gini impurity or information gain for classification tasks and mean squared error for regression tasks.

The tree continues to grow until a predefined stopping criterion is met, such as a specific depth or purity threshold. The resulting structure, with its branching decisions, forms a hierarchical tree. During prediction, new data traverses the tree, and the path followed determines the final outcome. Selecting the optimal depth of the decision tree involves tuning the hyperparameters using cross-validation and grid search in Python with the sci-kit learn library.

Decision trees offer transparency and ease of interpretation, making them valuable for extracting actionable insights. They handle both numerical and categorical data, and their resistance to outliers enhances robustness. The hierarchical structure of decision trees allows

for a step-by-step representation of decision-making processes, aiding in understanding complex relationships within the data.

However, these may be susceptible to overfitting, especially with deep trees, as they may capture noise in the training data. Additionally, decision trees might struggle to capture intricate patterns and non-linear relationships as effectively as more sophisticated models [15].

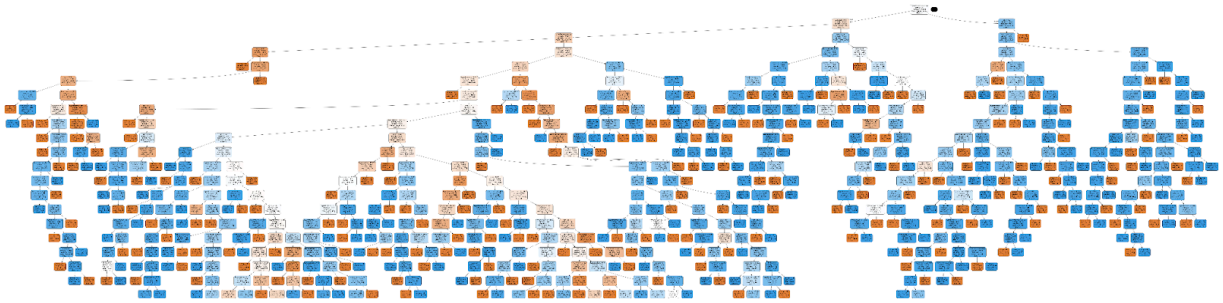

Figure 10: Decision Tree Produced in the Investigation

Random Forest Classifier

The Random Forest Classifier model is an ensemble ML model that is comprised of multiple decision trees, much like a forest. It creates diverse subsets of the training data using bootstrapped sampling, constructing individual trees with randomly chosen features to mitigate overfitting [16].

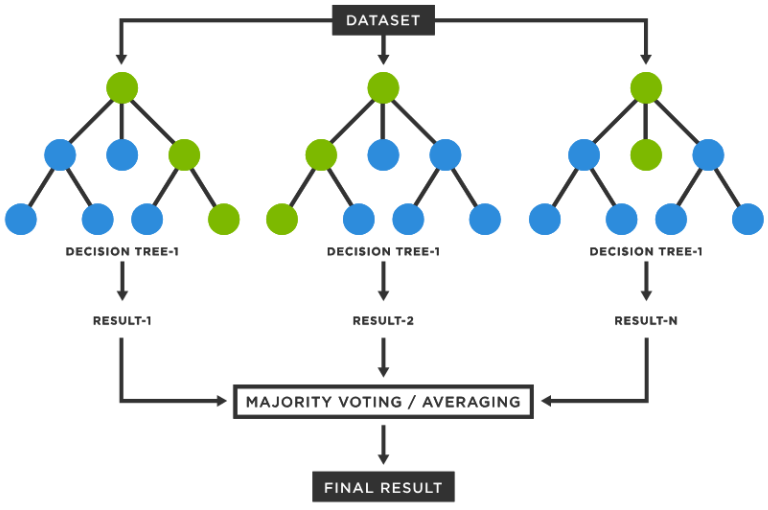

Figure 11: Random Forest Classifier Workflow (Gunay)

The process begins by randomly selecting subsets of the training data through a technique called bootstrapped sampling. For each subset, a decision tree is built independently, employing a top-down, recursive approach to identify the best feature splits at each node. The randomness extends to the selection of a random subset of features at each node, promoting diversity

among the trees. During prediction, the individual trees' outputs are aggregated through a majority voting mechanism for classification, producing a final prediction that is more robust and less prone to overfitting than a single decision tree. The random forest classifier, as an ensemble method, can be tuned to a certain number of estimators. It combines the predictions of all the estimators to produce a more accurate single prediction.

Typical decision trees are prone to overfitting the data set; however, the random forest classifier's design allows this model to compensate and generalize for unseen data by aggregating predictions from multiple trees. This model excels at handling high-dimensional data and provides a measure of feature importance, aiding in variable selection [17].

On the downside, random forests can be computationally expensive, particularly with a large number of trees in the forest. While they offer high predictive accuracy, the interpretability of the model decreases compared to individual decision trees.

KNN

K-Nearest Neighbors (KNN) is a versatile machine learning algorithm used for classification and regression. It assigns a data point to the majority class among its K nearest neighbors, determined by calculating distances in the feature space. The choice of "K" influences the number of neighbors considered for the decision. KNN is non-parametric, making no assumptions about data distribution, and relies on the entire dataset during both training and prediction.

In this investigation, the KNN model has been tuned. When iterated between 1 and 5 neighbors, performance was maximized at 1 neighbor.

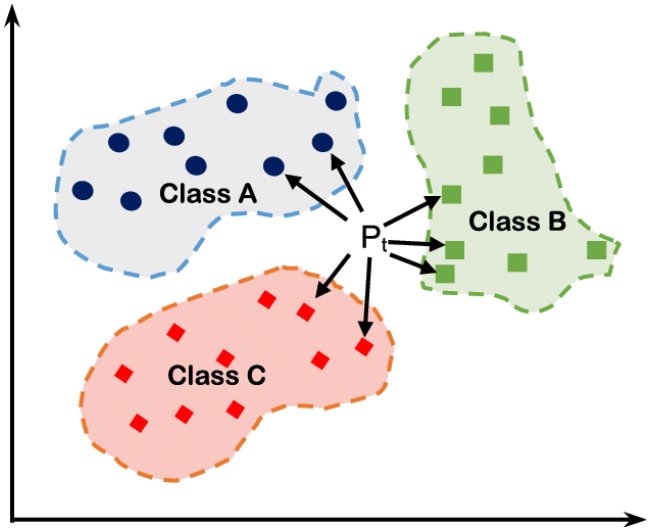

Figure 12: KNN Model (Sachinsony)

During training, the algorithm stores the entire dataset, creating a "neighborhood" for each data point. In the prediction phase, the algorithm identifies the K nearest neighbors of a given data point using a distance metric, commonly the Euclidean distance. For classification, the majority class among the K neighbors determines the predicted class. KNN's adaptability to local patterns makes it suitable for non-linear relationships.

KNN is an intuitive and easy-to-understand model, making it suitable for quick implementation and interpretation. It adapts well to local patterns and is non-parametric, making it effective for capturing non-linear relationships in the data.

However, KNN can be computationally expensive, especially with large datasets, as it requires distance calculations for each prediction. It is sensitive to irrelevant features and can struggle with imbalanced datasets where one class dominates [18].

SVM

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression tasks. The primary objective of SVM is to find a hyperplane in a high-dimensional space that best separates data points of different classes. In the context of binary classification, this hyperplane is the one that maximizes the margin, which is the distance between the hyperplane and the nearest data points (support vectors) of each class.

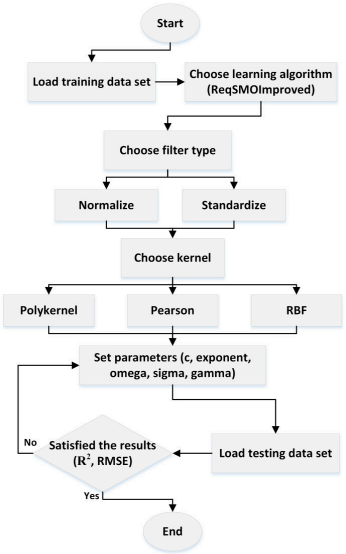

Figure 13: SVM Model Workflow (Alkhaled et al.)

SVM uses a kernel trick to map the input features into a higher-dimensional space, making it possible to find a hyperplane that separates the data in this transformed space. Common kernel functions include linear, polynomial, and radial basis functions (RBF). When utilizing Google Colab to run the code applied to this dataset, the radial basis function is the default kernel function. Additionally, when tested across the 3 aforementioned functions, the radial basis function returned the best results, hence it was used for this exploration.

For classification, once the hyperplane is determined, SVM assigns new data points to one of the two classes based on which side of the hyperplane they fall on.

SVMs excel in high-dimensional spaces and are versatile due to the availability of different kernel functions. They are effective in capturing complex decision boundaries and can handle cases where the relationship between predictors and the target is non-linear [19].

On the downside, SVMs might perform poorly with noisy or overlapping data. Tuning SVM hyperparameters, such as the choice of kernel and regularization parameters, can be challenging and may require careful consideration.

XGBoost

XGBoost is another ensemble ML method that combines the predictive power of multiple other models. Gradient boosting is the underlying principle behind the XGBoost model, whereby the model builds decision trees and uses errors from the previous trees to improve the model in successive iterations.

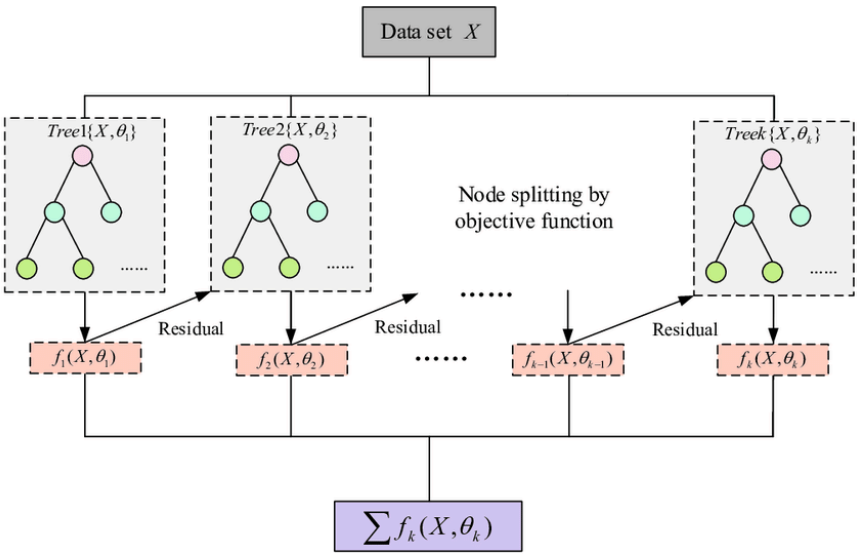

Figure 14: XGBoost Workflow (Guo et al.)

XGBoost works by iteratively building a series of decision trees, where each subsequent tree corrects the errors of the ensemble formed by the preceding trees. During training, XGBoost minimizes a specific loss function, incorporating both regularization terms and gradients of the loss with respect to the predictions. The algorithm employs a unique regularization term known as the "shrinkage" or "learning rate," controlling the contribution of each tree to the overall ensemble.

XGBoost also introduces a novel feature that incorporates second-order partial derivatives, enhancing its capability to capture complex relationships and interactions in the data. The ensemble is constructed by combining the predictions of all the trees, and the final model provides a robust and accurate prediction. It provides valuable insights into feature importance, handles missing data effectively, and is scalable, making it suitable for large datasets [20].

Yet, XGBoost demands careful tuning of hyperparameters to prevent overfitting, and its computational intensity might be a drawback, particularly for real-time applications or resource-constrained environments. The complexity of XGBoost may lead to overfitting on smaller datasets.

Performance Measurement

To evaluate the efficacy of the different machine learning models utilized, 5 performance measurements were employed: Accuracy, Precision, Recall, F1, and AUC scores. The formulas for calculating these measurements can be found in the appendix.

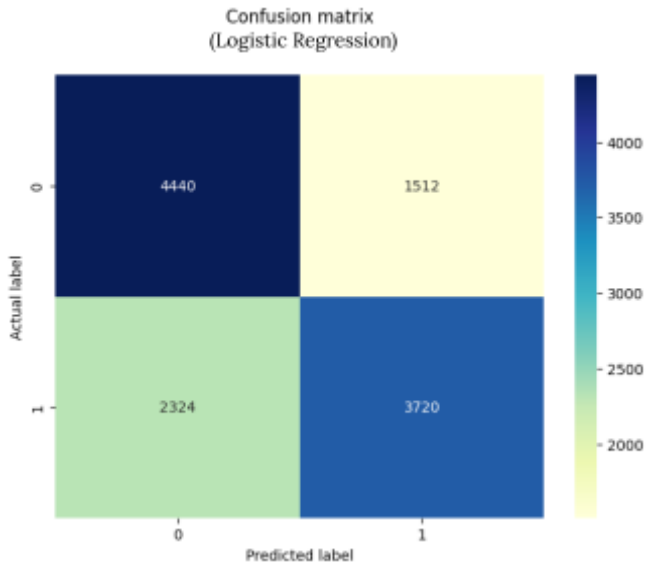

Figure 15: Logistic Regression Confusion Matrix

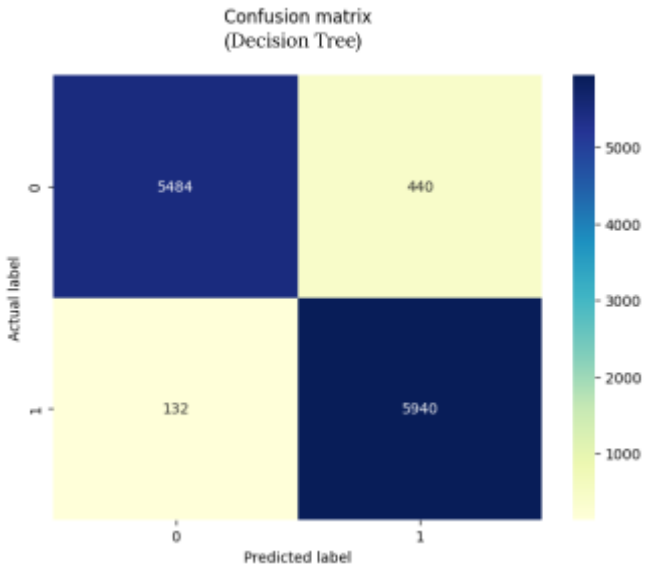

Figure 16: Decision Tree Confusion Matrix

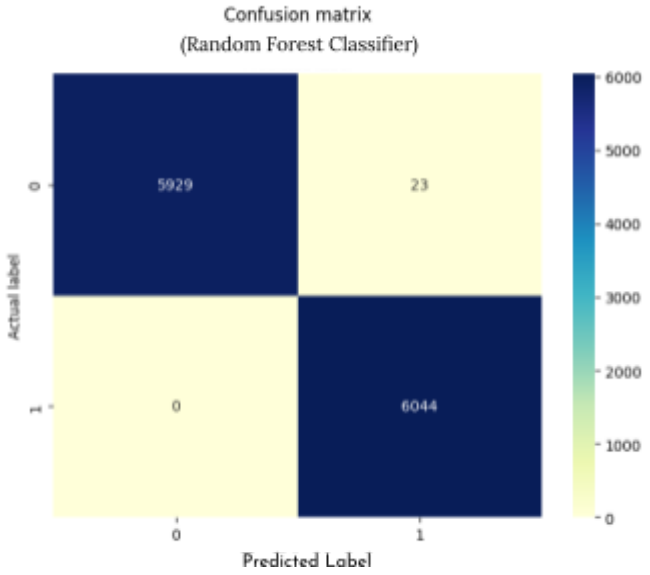

Figure 17: RFC Confusion Matrix

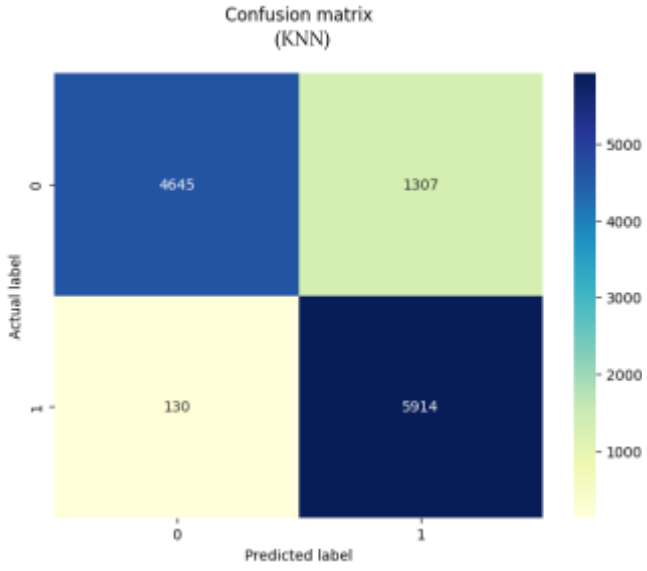

Figure 18: KNN Confusion Matrix

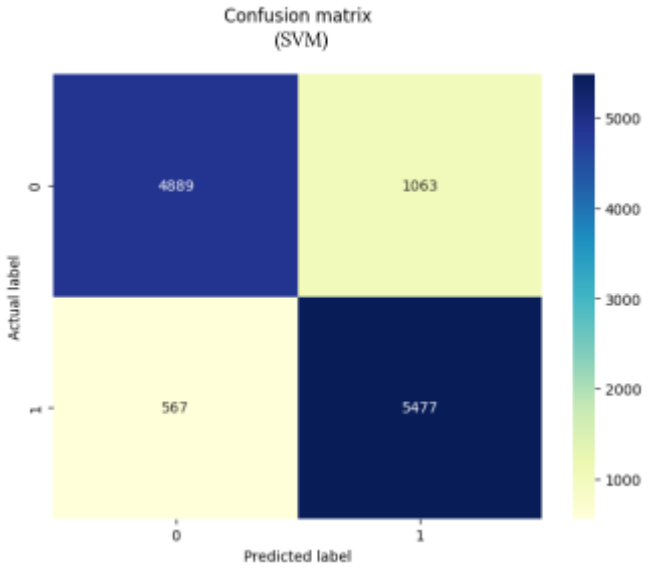

Figure 19: SVM Confusion Matrix

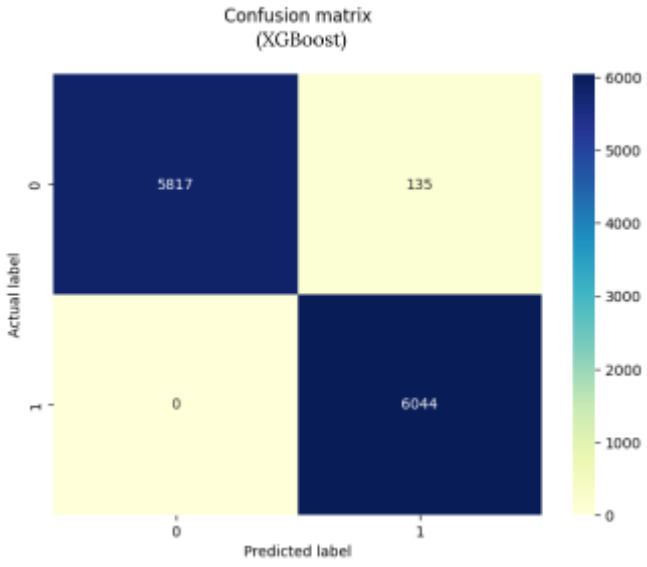

Figure 20: XGB Confusion Matrix

Results and Discussion

Classifiers	Accuracy	Precision	Recall	F1 Score	AUC
RFC	0.9981	0.9962	1.0000	0.9981	0.9981
XGBoost	0.9917	0.9837	1.0000	0.9918	0.9916
Decision Tree	0.9523	0.9310	0.9783	0.9541	0.9922
KNN (with 1 neighbor)	0.8802	0.8190	0.9785	0.8860	0.8795
SVM	0.8641	0.8375	0.9062	0.8705	0.8638
Logistic Regression	0.6802	0.7110	0.6155	0.9981	0.7443

Table 2: Performance Comparison

From Table 2, it is clear that all algorithms, barring logistic regression, have an acceptable level of accuracy. The random forest classifier is the most preferable model because of its higher level of accuracy, achieving 99.81% accuracy. It may be argued, however, that such a high level of accuracy may be attributed to the model overfitting. The accuracies were followed by the XGBoost (99.17%), Decision Tree (95.23%), KNN (88.02%), SVM (86.41%), and Logistic Regression (68.02%) models.

Limitations

Despite this investigation effectively demonstrating the extent to which different machine learning models can be applied in predicting the onset of strokes in patients, there are a few limitations to take into account.

The scope of the study may be constrained by the availability and quality of the data, as inherent biases or confounding factors within the data or analysis techniques could potentially influence the outcomes.

While certain classification models may demonstrate high accuracy, they often lack interpretability, making it difficult to understand how individual features contribute to the prediction outcomes. This "black-box" nature of some machine learning models means that, although we can observe the final prediction results, we cannot easily discern which features are most influential in tuning and optimizing the models. Consequently, the insights gained from these models are not fully transparent, which can be a significant drawback in medical and clinical applications where understanding the reasoning behind diagnosis is salient.

Conclusion

In conclusion, this paper has investigated the application of machine learning (ML) techniques for predicting stroke risk using comprehensive patient datasets. The study utilized various classifiers, including logistic regression, decision trees, support vector machines (SVM), and the Random Forest Classifier (RFC), to identify the most effective model for stroke prediction. Through rigorous experimentation and evaluation, the RFC emerged as the top-performing classifier, demonstrating superior accuracy in predicting stroke risk with an impressive 99.81% accuracy, with the XGBoost model closely behind at 99.17%.

Importantly, the methodologies and findings of this research are not confined to stroke prediction alone; they can be extended to other diseases and medical applications. By tweaking the procedure and adapting the features used for training, these machine learning models can be repurposed to predict a wide range of conditions, such as heart disease, diabetes, and certain types of cancer, and can even be tailored to forecast mental health disorders, such as depression and anxiety, by incorporating relevant behavioral and psychological parameters. Leveraging alternative and more thorough ML models such as Naïve Bayes, AdaBoost, Nearest Centroid, Voting Classifier, and multilayer perceptron may provide more conclusive results than the chosen subset of models in this study [21]. This expanded framework aims to boost both the reliability and overall performance of the predictive models.

Incorporating medical imaging data, such as CT scans, into the prediction models can provide valuable insights and improve classification accuracy by leveraging visual information beyond simple parameters. Image classification through computer vision is an underdeveloped niche industry that poses significant advancements to medical applications.

The findings of this study have significant implications for real-world healthcare applications. Identifying the best-performing ML model for stroke prediction can empower medical technologies to develop more accurate and reliable tools for detecting the onset of strokes. By leveraging the predictive capabilities of the RFC and other efficiently accurate machine learning models, healthcare providers can enhance early detection and intervention strategies, potentially reducing the incidence and severity of strokes and improving patient outcomes [22].

This study extends beyond its immediate findings in that showcasing the efficacy of ML techniques in stroke prediction underscores the importance of incorporating data-driven approaches into clinical decision-making processes. The adoption of advanced predictive analytics in healthcare can revolutionize patient care, facilitating proactive interventions, personalized treatments, and improved patient management strategies [23]. In an ideally utilized situation, this would save the lives of many diagnosed patients and ensure they do not suffer any fatal or radical consequences.

References

[1] World Health Organization, “World Stroke Day 2022,” WHO, Oct. 29, 2022. [Online]. Available: https://www.who.int/srilanka/news/detail/29-10-2022-world-stroke-day-2022

[2] E. M. Alanazi et al., “Predicting Risk of Stroke from Lab Tests Using Machine Learning Algorithms: Development and Evaluation of Prediction models,” JMIR Form. Res., vol. 5, no. 12, p. e23440, Dec. 2021, doi: 10.2196/23440.

[3] D. Atallah, M. Badawy, A. El-Sayed, and M. Ghoneim, “Predicting Kidney Transplantation Outcome Based on Hybrid Feature Selection and KNN Classifier,” Multimedia Tools Appl., vol. 78, pp. 20383–20407, 2019, doi: 10.1007/s11042-019-7370-5.

[4] “How Many People are Affected by/at Risk for Stroke?,” Eunice Kennedy Shriver National Institute of Child Health and Human Development, U.S. Department of Health and Human Services. [Online]. Available: https://www.nichd.nih.gov/health/topics/stroke/conditioninfo/risk [Accessed: Jul. 31, 2024].

[5] World Health Organization, “Stroke, Cerebrovascular Accident,” WHO, 2024. [Online]. Available: https://www.emro.who.int/health-topics/stroke-cerebrovascular-accident/index.html

[6] Stroke Awareness Foundation, “Stroke Facts & Statistics,” Jul. 11, 2023. [Online]. Available: https://www.strokeinfo.org/stroke-facts-statistics/

[7] G. G. Sailasya and G. L. A. Kumari, “Analyzing the Performance of Stroke Prediction using ML Classification Algorithms,” Int. J. Adv. Comput. Sci. Appl., vol. 12, 2021. [Online]. Available: https://thesai.org/Downloads/Volume12No7/Paper_65-Analyzing_the_Performance_of_Stroke_Prediction.pdf [Accessed: Jul. 31, 2024].

[8] E. Dritsas and M. Trigka, “Stroke Risk Prediction with Machine Learning Techniques,” Sensors, vol. 22, no. 13, p. 4670, Jun. 2022, doi: 10.3390/s22134670.

[9] R. Islam, S. Debnath, and T. Islam, “Predictive Analysis for Risk of Stroke using Machine Learning Techniques,” in Proc. IC4ME, 2021, pp. 1–4, doi: 10.1109/IC4ME253898.2021.9768524.

[10] A. Suresh, “What is a confusion matrix?,” Medium, Analytics Vidhya, Jan. 18, 2024. [Online]. Available: https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5

[11] “Logistic Regression in Machine Learning - Javatpoint,” Javatpoint. [Online]. Available: https://www.javatpoint.com/logistic-regression-in-machine-learning [Accessed: Dec. 11, 2023].

[12] M. Song, “Logistic regression explained,” Medium, Sep. 24, 2023. [Online]. Available: https://medium.com/@msong507/logistic-regression-explained-2d1b8babe6c1

[13] “Decision Tree Algorithm in Machine Learning - Javatpoint,” Javatpoint. [Online]. Available: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm [Accessed: Dec. 11, 2023].

[14] C. V. Nicholson, “Decision tree,” Pathmind. [Online]. Available: https://wiki.pathmind.com/decision-tree [Accessed: Jul. 30, 2024].

[15] M. Attard, “8 Key Advantages and Disadvantages of Decision Trees,” Inside Learning Machines, May 1, 2024. [Online]. Available: https://insidelearningmachines.com/advantages_and_disadvantages_of_decision_trees/#2_Robust_to_Outliers

[16] D. Gunay, “Random forest,” Medium, Sep. 14, 2023. [Online]. Available: https://medium.com/@denizgunay/random-forest-af5bde5d7e1e

[17] “Demystifying the Random Forest Algorithm for Accurate Predictions,” Spotfire. [Online]. Available: https://www.spotfire.com/glossary/what-is-a-random-forest [Accessed: Dec. 11, 2023].

[18] Sachinsoni, “K Nearest Neighbours—Introduction to Machine Learning Algorithms,” Medium, Jun. 11, 2023. [Online]. Available: https://medium.com/@sachinsoni600517/k-nearest-neighbours-introduction-to-machine-learning-algorithms-9dbc9d9fb3b2

[19] A. Alkhaled, A. Kabutey, K. Selvi, Č. Mizera, P. Hrabě, and D. Herak, “Application of Computational Intelligence in Describing the Drying Kinetics of Persimmon Fruit (Diospyros kaki) During Vacuum and Hot Air Drying Process,” Processes, vol. 8, p. 544, 2020, doi: 10.3390/pr8050544.

[20] R. Guo, Z. Zhao, T. Wang, G. Liu, J. Zhao, and D. Gao, “Degradation State Recognition of Piston Pump Based on ICEEMDAN and XGBoost,” Appl. Sci., vol. 10, p. 6593, 2020, doi: 10.3390/app10186593.

[21] M. Singh, S. Verma, and P. Singhal, “A Comparative Study of Stroke Prediction Algorithms using Machine Learning,” in Advances in Intelligent Systems and Computing, 2023, doi: 10.1007/978-3-031-35641-4_22.

[22] A. Tazin et al., “Stroke Disease Detection and Prediction using Robust Learning Approaches,” J. Healthc. Eng., Nov. 26, 2021, doi: https://doi.org/10.1155/2021/7633381.

[23] M. Wiryaseputra, “Stroke Prediction using Machine Learning Classification Algorithm,” 2022. [Online]. Available: https://www.researchgate.net/publication/362175348_Stroke_Prediction_Using Machine_Learning_Classification_Algorithm

Appendix

A.1 $$Accuracy = \frac{True\ Negative + True\ Positive}{True\ Negative + False\ Negative + True\ Positive + False\ Positive}$$

A.2 $$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

A.3 $$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

A.4 $$F1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

A.5 AUC Value: $$ROC - AUC = \int_0^1 TPR(FPR)dFPR$$

A.6 $$TPR = \frac{TP}{TP + FN}$$ (TPR = true positive rate)

A.7 $$FPR = \frac{FP}{FP + TN}$$ (FPR = false positive rate)