
Jyotir Moy Chatterjee • Shailendra K. Saxena
Editors

Artificial Intelligence in Medical Virology

Stroke Disease Prediction Model Using ANOVA with Classification Algorithms

8

Sulaiman Olaniyi Abdulsalam, Micheal Olaolu Arowolo,
and Oroghi Ruth

Abstract

Stroke is a health ailment where the brain plasma blood vessel is ruptured, triggering impairment to the brain. Symptoms may appear when the brain's blood flow and other nutrients are disrupted. Stroke is the leading cause of bereavement and disability universally, according to the World Health Organization. To predict a patient's risk of having stroke, this project used machine learning (ML) approach on a stroke dataset obtained from Kaggle, the ANOVA (Analysis of Variance) feature selection method with and without the following four Classification procedures; Logistic Regression, K-Nearest Neighbor, Naïve Bayes, and Decision Tree, after which the dataset was preprocessed. The K-Nearest Neighbor algorithm gave the best performance accuracy of approximately 97% without ANOVA and Decision Tree algorithm with ANOVA method gave 96%. The accuracy of the developed models employed in this study is substantially better, showing that the models employed in this study are much more reliable. And it can be deduced from previous existing works.

Keywords

ANOVA · Stroke · KNN · Decision tree · Machine learning

S. O. Abdulsalam · O. Ruth

Department of Computer Science, Kwara State University, Malete, Nigeria

M. O. Arowolo (✉)

Department of Computer Science, Landmark University, Omu-Aran, Nigeria

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

J. M. Chatterjee, S. K. Saxena (eds.), *Artificial Intelligence in Medical Virology*, Medical Virology: From Pathogenesis to Disease Control, https://doi.org/10.1007/978-981-99-0369-6_8

8.1 Background of the Study

The health sector has played an important role by discovery various set of disease that leads to patient death, and Stroke has been shorted has one of the deadliest diseases in the world. Stroke is a health disorder where the brain's blood source is damaged, causing cell death. Strokes can be divided into ischemic and hemorrhagic which are strokes recognized by doctors today. When a blood clot lodges in the brain, it causes an ischemic stroke, which is more common. This reduces blood supply to certain parts of the brain. A hemorrhagic stroke arises when the brain's blood vessel bursts. Blood builds up as a result of this. The location of the stroke in the brain and the number of brain cells injured are strongly associated to the degree of the stroke (Donkor 2018).

Because stroke is such a complicated medical illness, predicting the signs of a stroke is extremely difficult. Doctors, on the other hand, can manually assess a patient's risk of ischemic stroke grounded on the sternness of their metabolic condition, a group of diseases that comprises high blood pressure, excess body fat around the abdomen, and abnormal cholesterol levels (Liu and Cong 2017).

It has attracted the attention of many in the technology sector to use machine learning techniques to efficiently identify strokes by routinely gathering datasets and offering reliable diagnosis findings. The most widely used algorithms include Logistic Regression, Support Vector Machines, C4.5, Neural Networks, and others (Bandi et al. 2020).

Problems encountered while using three distinct algorithms such as Decision Trees, Naive Bayes, Neural Networks algorithms show that Decision Tree gave an accuracy of 75%. Regarding the outcomes obtained, the model could not be useful for real cases (Nwosu et al. 2019).

To predict stroke, this study proposes a machine learning technique using ANOVA (Analysis of Variance) method for feature selection with classification algorithms; logistic regression, K-Nearest Neighbors, Naive Bayes, and Decision Tree algorithm.

8.2 Related Works

A data mining approach for predicting ischemic stroke using classification and logistic regression was suggested, by looking into the causes of ischemic stroke. The data was then preprocessed, cleaned, and analyzed using the data package WEKA (Waikato Environment for Knowledge Analysis) 3.6 and C4.5 algorithm, as well as logistic regression. They analyzed the sample data they had collected using Microsoft "XLSTAT." It was discovered that in their case study, the logistic regression model allowed them to examine the relationship between the occurrence of ischemic stroke and its risk variables. The outcome obtained a high sensitivity of 78% (Abdelwahab and Khalid 2015).

According to Kansadub (2015), three distinct algorithms were used to forecast the likelihood of a stroke. Naïve Bayes, Decision Trees, and Neural Networks were the algorithms used. The Decision tree obtained an accuracy of about 75%.

Arslan et al. (2016) proposed an SVM, Stochastic Gradient Boosting, then Logistic Regression approach for predicting ischemic stroke from Turgut Ozal Medical Center, which contained 112 healthy people and 80 patients record with two target variables. They had to use a resampling method with a 10-fold cross validation. In comparison to other models, Support Vector Machines generated the highest results for predicting ischemic stroke, with 95% accuracy.

A team from a Taiwanese medical university created a model to systematize the primary identification of ischemic stroke (Chiun-Li-Chin et al. 2017). For this model, they employed a convolutional neural network deep learning technique. They used computed tomography (CT) scans of the brains to look for signs of stroke. The computed tomography scans were preprocessed by the algorithm to remove the impossible area where a stroke could occur. Then they selected the reinforcement imageries and used information augmentation technique to enhance the amount of reinforcement imageries. They then used patch photos as input to train and test the CNN. They employed 256 patch images for the training and testing, which recognized the ischemic stroke, the result obtained 90% accuracy.

Aishwarya (2018) carried out a survey on AI applications in stroke for accurately predicting the onset of stroke. Several prediction algorithms and parameters that include patient's characteristics like gender, age, height, BMW (Bayerische Motoren Werke AG), etc., were proposed. They also developed a data model utilizing decision tree method to assess these limitations. The result was evaluated and obtained an accuracy of 95%. To use it, they created a training model that aided in the comparison of recent survey data.

The authors of Ohoud (2018) used various classification approaches to forecast the likelihood of stroke in human. The data was gathered from Saudi Arabia's Health Hospitals. C4.5, multi-layer perceptron JRip classification algorithms were adopted. The model attained a 95% accuracy, the period for training then prediction is lengthier because of a hybridized complicated procedure.

Nwosu (2019) investigated on stroke prediction for training Random Forest, Decision Tree, and Multi-layer Perceptron. The accuracies acquired for the three techniques were quite comparable, with only minor variations. Decision Tree calculated an accuracy of 74%, Random Forest had 75%, and Multi-layer Perceptron had a measured accuracy of 75.02%. According to this manuscript, the Multi-layer Perceptron is more precise than other techniques. The accuracy measure used solely for calculating the performance, which may not constantly produce positive outcomes.

Build a model to predict the long-term effects of ischemic stroke (JoonNyung et al. 2019). They observed machine learning algorithms to envisage ischemic stroke outcomes. A prospective group of individuals with ischemic stroke to conduct a historical investigation. Deep neural networks, random forests, and logistic regression were constructed as machine learning models. They compared all of their predictions after that. Researchers also compared these models to an acute stroke

registry and a Lausanne score analysis to see how accurate they were. They also included 2604 patients and 78% of them had promising outcomes. The AUC curve of the deep neural network model was greater than the ASTRAL score; however, the AUC curves of the logistic regression and random forest models were not advanced than the ASTRAL score. Deep neural networks outperformed all other models, according to the results. It was more suited to forecasting outcomes.

Luis (2019) investigated the theory that advanced machine learning-based model methods could be used to test the hypothesis. They investigated that semi-monitoring methods could aid on the analysis of stroke type. These tools can even be used to predict future risks, such as the patient's eventual death. They collected dataset composed of the medical records. They analyzed over six distinct measures using seven different machine learning algorithms. Random Forest models outperformed all other algorithms in terms of stroke prediction and death prediction, with values 97%.

Alotaibi (2019) proposed a machine learning model for predicting stroke. They built the model using a variability of machine learning methods such as Decision Trees, Naïve Bayes, and Support Vector Machines, and then compared their results. With the algorithms they employed, researchers were able to achieve an accuracy of 60%.

Singh (2020) proposed a machine learning technique for predicting stroke dataset, C4.5 Decision Tree algorithm, Artificial Neural Networks, Principal Component Analysis, and Support Vector Machine were used to identify the best answer.

People's social media posts were used to forecast strokes (Pradeepa et al. 2020) by employing the Detecting Risk Factors of Stroke (DRFS) method to recognize the several indications related with stroke ailment in this study. Natural Language Processing is being used to excerpt text from social media posts, which increases the model's total implementation period, which is undesirable.

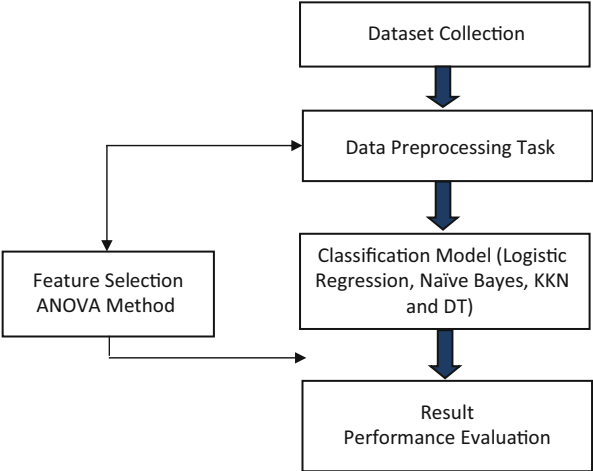
Bandi et al. (2020), based on the authors, various machine learning procedures for instance Decision Tree, Naïve baye, Logistic Regression, Linear SVM, Poly SVM, Random Forest, Adaboost, and Adaboost with SGD. Random Forest Classification performs best with an accuracy of 96.97% after using it stroke prediction model (SPR) and the error rate is 0.03%.

Gangavarapu (2021) proposed a machine learning model for predicting stroke for multiple biological features. Naïve Bayes outperformed with an accuracy of 82%. This study extended by training the model with Neural Networks.

8.3 Materials and Methods

The proposed model of this project will use ANOVA method as a feature selection method with logistic regression, K-Nearest Neighbors, Naive Bayes, and Decision Tree algorithms for stroke classification. The methods to be used in this model research are split by the main phases of the development process, to formulate the problem, data set collection, prepare the data, build the model, implement the model and then conclude the results. It is important to train the data using various

Fig. 8.1 Proposed model workflow



algorithms and then compare selected features with trained data. To extract the patterns from the dataset, and then create a model based on the designs extracted. The trained dataset is supplied and then trained based on the dataset. After the testing operation, the model predicts whether or not the pictures are for stroke or otherwise. The result is then evaluated based on this flow (Fig 8.1)

8.3.1 Classification

Unstructured and structured data can be classified using the classification method. In the first step of the technique, the class of data points is predicted using the information provided. The classes are described using the phrases target, label, and categories. Replicating the transformation matrix from discrete input parameters to discrete independent variables is the goal of predictive analysis. The primary objective is to identify which data classification the new information falls under (Sina Mirabdolbaghi and Amiri 2022).

Logistic Regression What Exactly Are They, and How Do They Plan to Operate? In order to make an accurate prediction regarding the likelihood of a target variable, the logistic regression supervised learning detection model is utilized. Because the nature of the aim or response variables is binary, there are only two classes available to choose from. To put it another way, the dependent variable is binary, meaning that the data can either be written as 1 (which stands for “success” or “yes”) or 0 (which stands for “failure” or “no”).

Naive Bayes Algorithm is a method of categorization that uses the Bayes Theorem as its foundation and operates on the assumption that all variables are distinct of each

other. In essence, it is a sophisticated probabilistic machine learning classification technique. It is simple to set up and can handle massive datasets.

The Bayes theorem is a scientific principle that evaluates the chance of an occurrence depending on certain antecedent conditions being met. Bayes' theorem was named after the mathematician Thomas Bayes. It is a tool for estimating the conditional probability by taking into consideration the Likely, the Category Preceding Probabilistic, and the Coefficient of determination Preceding Potential.

K-Nearest Neighbor This is a simple approach that stores all previously collected examples and employs a similarity metric to classify any additional data or cases that are gathered. Its most common application is to determine the categorization of a dataset based on the classification of its immediate surroundings (El Houari 2019).

Decision Tree The collection of supervised learning algorithms includes the Decision Tree method as one of its own members. In contrast to other methods of supervised learning, the decision tree technique is able not just for the purpose of solving regression difficulties but also classification issues. The purpose of utilizing a Decision Tree is to construct a learning model that might be used to perform classification or quantity of the target attribute by adopting simple decision principles deduced from previous data. This can be accomplished through the usage of the Decision Tree (training data) (KDnuggets 2020).

Performance Metrics

A structure known as the confusion matrix is one that is utilized in the process of evaluating the efficacy of prediction model for a specific instance of test data. It is impossible to determine without having knowledge of the actual outcomes of the test results. Understanding the matrices is not difficult, but the terminology that goes along with it could be difficult. Because it presents the mistakes in the model's efficiency in the form of a matrix, an error matrix has earned the moniker "error matrix" (Javatpoint 2021).

- i. True Negative: Both the model and the true or actual number made the same prediction, which was that it would be negative.
- ii. True Positive: The model anticipated a positive outcome, and the observed data substantiated that expectation.
- iii. False Negative: A Type-II error occurs when the model predicts that the response will be no, while the actual answer will be yes.
- iv. False Positive: The model anticipated that the choice is obvious, however the outcome turned out to be no. Some other term for this kind of error is a Type-I error.

Accuracy When attempting to determine the accuracy of classification tasks, one of the most significant characteristics to take into consideration is the accuracy of the problems themselves. It identifies the percentage of times that the model accurately forecasts the result. The proportion of right predictions produced by the classification

to the overall amount of values predicted by the classifiers is something that may be calculated. The formula can be broken down as follows:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}(\text{Tp} + \text{Tn})}{\text{Total population}(\text{Tp} + \text{Fp} + \text{Fn} + \text{Tn})}$$

Recall It is the sum of all of the positive classifications that can be attributed to the model's accurate predictions. It is essential that the rate of recall be as large as is practically possible.

$$\text{Recall} = \frac{\sum \text{Tp}}{(\sum \text{Tp} + \sum \text{Fn})}$$

Specificity measured by the number of false positives that our ML model has produced. By using the following formula in conjunction with the confusion matrix, we are able to simply calculate it.

$$\text{PPV} = \text{TP}/(\text{TP} + \text{FP})$$

Precision It can be measured as the proportion of valid outcomes generated by the model or can be characterized as the proportion of all attracted significant that the model accurately anticipated to be true. It can be determined by applying the formula that is as follows:

$$\text{Precision} = \frac{\sum \text{Tp}}{\sum \text{Tp} + \sum \text{Fp}}$$

F1 Score When comparing two models, it is problematic to do so when one has low precision but a high recall while the other has the opposite. Therefore, we are able to make use of the F-score for this objective. With this score, we are able to evaluate both the recall and the precision at the same time. When the recall and precision are equal, the F-score is at its highest possible level.

$$\text{F1 - Score} = 2\text{TP}/(2\text{TP} + \text{FP} + \text{FN})$$

Matthews's Correlation Coefficient The quality of binary and multiclass classifications can be evaluated with its help through the process of machine learning. It is usually recognized as a balanced measure that can be employed even when the classes are of significantly different sizes since it takes into consideration both genuine and false positives and negatives and it accounts for both true and false positives and negatives. The MCC is, in essence, a correlation coefficient value that can take on any value from minus one to plus one. A prediction that is perfect is denoted by a coefficient value of +1, while a prediction that is random and average is denoted by a coefficient value of 0, and a prediction that is inverse is denoted by a

coefficient value of -1 . The statistic is sometimes referred to by another term, which is the phi coefficient.

$$\text{MCC} = \frac{\text{TP} * \text{TN} - \text{FP} * \text{FN}}{\sqrt{(\text{TP} + \text{FP}) * (\text{TP} + \text{FN}) * (\text{TN} + \text{FP}) * (\text{TN} + \text{FN})}}$$

Python, an open-source programming language, and the Jupyter Notebook integrated development environment (IDE) were used to carry out the study (Integrated Development Environment). Python tool, 4 gigabytes of random access memory, 20 gigabytes of hard drive space, Windows 10 operating system, and Intel (R) Pentium(R) processor are all included.

8.4 Results and Discussions

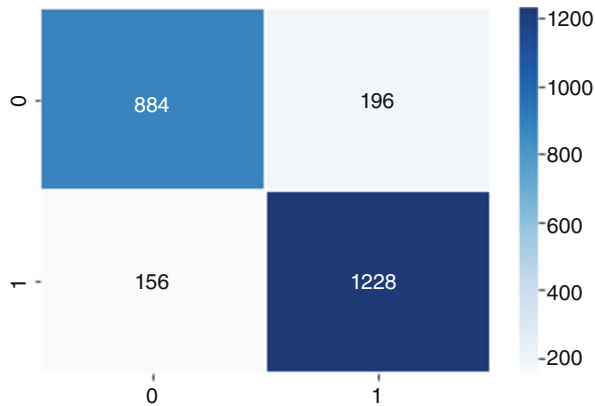
This study explains how the implementation was carried out and evaluated. This also includes a detailed description of the conclusions and results of the proposed model. This result demonstrates that the study was completed in accordance with the study's goals and objectives. The stroke prediction dataset was made available by [Kaggle.com](https://www.kaggle.com). The rows in this dataset total 5110, and there are 12 columns. The primary attributes that are contained inside the columns are as follows: id, gender, age, hypertension, heart disease, ever married, work type, residence type, average glucose level, body mass index, smoking status, and stroke. The value of the output column labeled "stroke" is either "1" or "0." If the value is "0," then there are no chances of causing a stroke, but if the value is "1," then there is a chance of having a stroke. A severely unbalanced dataset is the consequence of the likelihood of a "0" in the output column ("stroke") not being proportional to the probability of a "1" in the same column. This is due to the fact that the probabilities are not equal. In the stroke column, there are a total of 249 rows with the value "1," while the remaining 4861 rows have the value "0." In order to achieve greater precision, the data are checked for consistency via preprocessing beforehand.

It is necessary to perform preprocessing before to the development of a model in order to eliminate distortion and anomalies from the dataset, both of which have the potential to deviate from good training. At this point, we address anything about the model that may be preventing it from functioning more effectively.

To complete this task, the Jupyter Environment is used for coding, and a Python interpreter is used to run the code with the loaded dataset in "csv" file format, which contains all attributes with null values. In order for the machine to understand the string literals (objects) in the dataset, the label encrypting function alters the attribute with the string literal to integer values, which the model can use for prediction because computers are normally trained in numbers.

The model is constructed once the preprocessing phase has been finished in its entirety. To increase efficiency and accuracy for this assignment, the dataset has been partitioned into training data and testing data, with 70% of the dataset dedicated to training data and 30% dedicated to testing data. In order to train the model,

Fig. 8.2 Logistic Regression classification (TP = 1228, TN = 884, FP = 196, FN = 156)



multiple classification algorithms, such as logistic regression, Naive Bayes, and the K-Nearest Neighbors, and Decision Tree algorithms, as well as the AVONA feature selection algorithm, are utilized after the dataset has been partitioned and the data has been normalized.

The dataset that is utilized for stroke predictions is represented in a manner that is highly imbalanced. There are 5110 rows in the dataset, with 249 rows showing the probability of a stroke and 4861 rows representing the occurrence of a stroke. Although such data can be utilized to train techniques, other accuracy measures such as precision and recall are poor due to the nature of the data. If the handling of unbalanced data is not done correctly, the results will be wrong, and the forecast will not be accurate either. As a consequence of this, the imbalanced data must be handled before one can build a strategy that is efficient. SMOTE and Edited Nearest Neighbors, also known as hybrid ways to tackle imbalanced datasets, were used to balance the dataset. The method employs both over-sampling and under-sampling to balance both of the divided datasets. Hybrid strategies are commonly used to address imbalanced datasets (X, y).

On the stroke dataset, the prediction was made by omitting the ANOVA feature selection approach and making use of four classification methods. These classification algorithms included Logistic Regression, Naïve Bayes, K-Nearest Neighbors, and Decision Tree algorithms. Nevertheless, a description of each procedure may be found in the confusion matrix that is shown in the pictures below. The confusion matrix used by the Logistic Regression algorithm can be found depicted in Fig. 8.2.

The confusion matrix for the Naive Bayes method is shown in Fig. 8.3.

The confusion matrix for the K-Nearest Neighbor algorithm is shown in Fig. 8.4.

The confusion matrix for the Decision Tree method is shown in Fig. 8.5.

The stroke dataset was used to make the prediction, along with the ANOVA feature selection approach and four classification methods including Logistic Regression, Naïve Bayes, K-Nearest Neighbors, and Decision Tree algorithms. These algorithms were used. On the other hand, the confusion matrix is shown in

Fig. 8.3 Naïve Bayes classifier (TP = 1218, TN = 869, FP = 211, FN = 166)

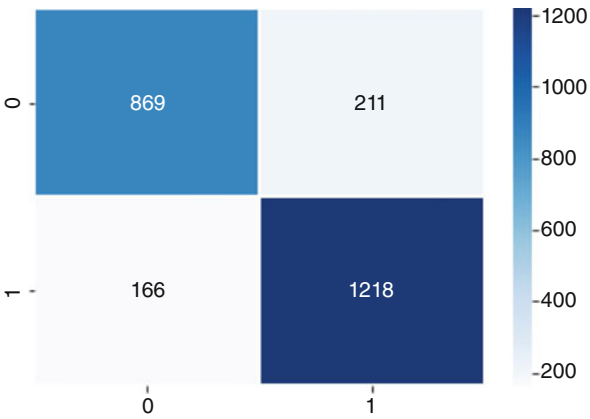


Fig. 8.4 K-Nearest Neighbor classifier (TP = 1381, TN = 1021, FP = 59, FN = 3)

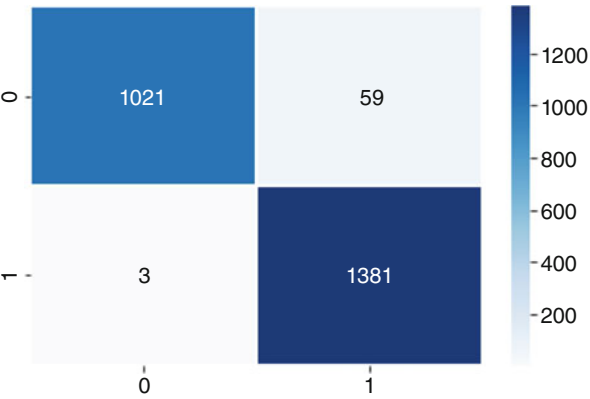


Fig. 8.5 Decision Tree classifier (TP = 1331, TN = 1007, FP = 73, FN = 53)

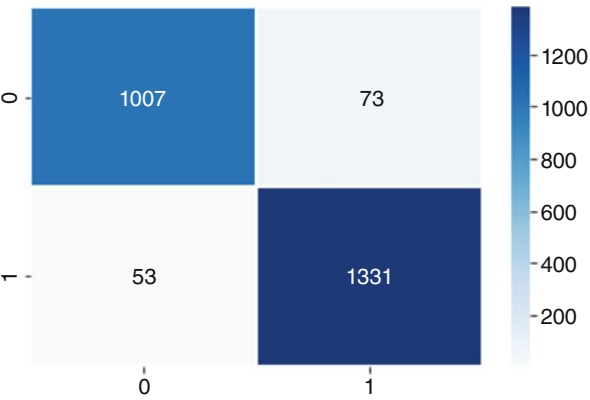


Fig. 8.6 Logistic Regression classification (TP = 1237, TN = 874, FP = 194, FN = 154)

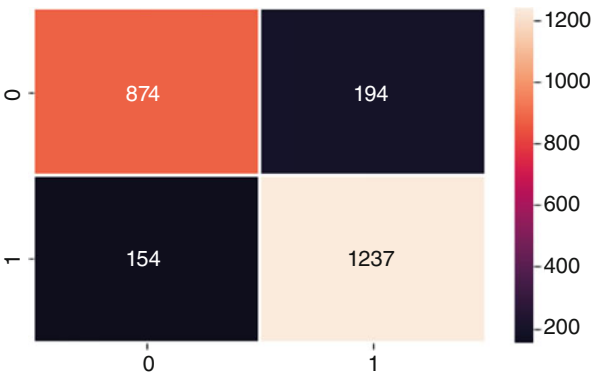


Fig. 8.7 Naïve Bayes classifier (TP = 1238, TN = 849, FP = 219, FN = 153)

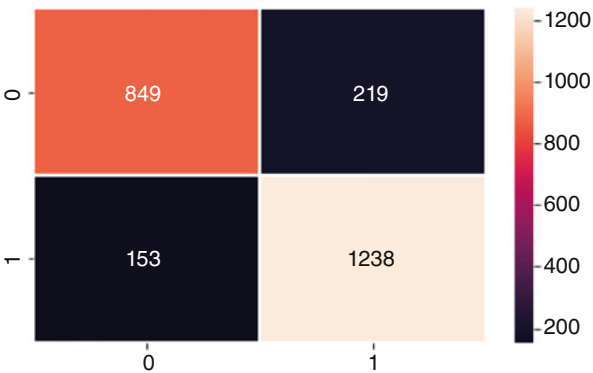
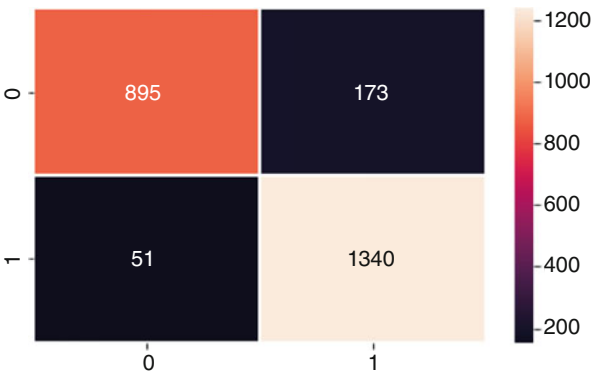


Fig. 8.8 K-Nearest Neighbor classifier (TP = 1340, TN = 895, FP = 173, FN = 51)



the following pictures, which describe each algorithm. In Fig. 8.6, you can see the confusion matrix that corresponds to the Logistic Regression algorithm.

- The confusion matrix for the Naive Bayes method is shown in Fig. 8.7.
- The confusion matrix for the K-Nearest Neighbor algorithm is shown in Fig. 8.8:
- The confusion matrix for the Decision Tree method is shown in Fig. 8.9.

Fig. 8.9 Decision Tree classifier (TP = 1350, TN = 1002, FP = 66, FN = 41)

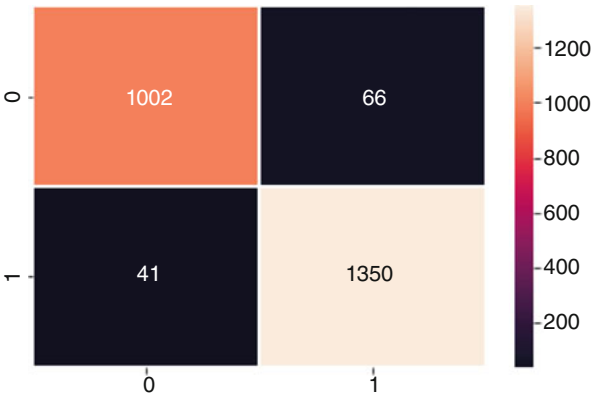


Table 8.1 Calculation and comparison of performance

Performance	Logistic Regression	ANOVA + Logistic Regression	Naïve Bayes	ANOVA + Naïve Bayes
Sensitivity	88.73	88.93	88.01	89.00
Specificity	81.85	81.84	80.46	79.49
Precision	86.24	86.44	85.23	84.97
Accuracy	85.71	85.85	84.70	84.87
F1 score	87.46	87.67	86.60	86.94
Mathews correlation coefficient	70.91	71.11	0.68.83	69.09

Table 8.2 Calculation and comparison of performance

Performance	K-Nearest Neighbor	ANOVA + K-Nearest Neighbor	Decision Tree	ANOVA + Decision Tree
Sensitivity	99.78	96.33	96.17	97.05
Specificity	94.54	83.80	93.24	93.82
Precision	95.90	88.57	94.80	95.34
Accuracy	97.48	90.89	94.89	95.65
F1 score	97.80	92.29	95.48	96.16
Mathews correlation coefficient	94.96	81.64	89.61	91.14

In this study, classification was accomplished through the use of logistic regression, the Naïve Bayes algorithm, the K-Nearest Neighbors algorithm, and the Decision Tree algorithm. The data was passed into the aforementioned classifiers both with and without the application of the ANOVA feature selection method.

Performance indicators such as confusion matrix, sensitivity (Recall), specificity, precision, accuracy, F1 score, and Mathew's correlation coefficient were utilized in order to analyze the results of the acquired confusion matrices. The performance measures of each classifier are displayed in Tables 8.1 and 8.2, respectively, both with and without the ANOVA approach.

Table 8.3 Comparison of the obtained results with existing works

Authors	Algorithms/methods used	Accuracy (%)
Arslan et al. (2016)	Stochastic gradient boosting + SVM	95
Chiun-Li-Chin (2017)	Convolutional neural network	90
Ohoud Almadani (2018)	C4.5 + JRip + multi-layer perceptron	95
Alotaibi (2019)	Decision trees, Naïve Bayes, support vector machines	60
Bandi et al. (2020)	Random Forest	96.97
Gangavarapu Sailasya (2021)	Naive Bayes	82

Table 8.3 compares the algorithms used in this study with previous work result been done.

8.5 Conclusion

Due to the multiple symptoms associated with the Stroke condition, it is difficult for medical practitioners to anticipate whether a patient is likely to have Stroke or not. This study used a machine learning approach to create a system that allows doctors to predict if a patient is going to suffer a stroke or not. The project used classification techniques to predict outcomes, with four various classifier algorithms (Logistic Regression, Naïve Bayes, K-Nearest Neighbor, and Decision Tree algorithm, with and without ANOVA feature selection method). The Algorithms were trained using a dataset obtained from the Kaggle website, which contained 5110 rows and 12 columns of data collected from various hospitals, and the following performance metrics were used to determine how well the algorithm performed: confusion matrix accuracy, precision, sensitivity, specificity, and Mathews Correlation Coefficient. The K-Nearest Neighbor algorithm without ANOVA method delivers the best accuracy, with approximately 97% and Decision Tree with ANOVA method gave 96%. In conclusion, the Decision Tree with ANOVA method is said to have performed more effective than the other algorithms employed in this project for solving the problem due to the fact that classifiers predict accurately when combined with feature selection algorithm. It is important to note that the Stroke Prediction System was created to help the medical industry. In the future, this study recommend that the system should be developed using web app and mobile app to enable doctors to interact with the diagnosed patient for easily, other algorithms should be used and also evaluated using real-life patient data to learn how effectively the system performs.

Appendix

```

import numpy as np
import pandas as pd
from pandas import Series, DataFrame
import matplotlib.pyplot as plt
from scipy import stats
import seaborn as sns

stroke_data.head()
stroke_data = pd.read_csv("stroke predict.csv")

stroke_data.info()
stroke_data.isnull()
stroke_data.describe()

stroke_data = stroke_data.fillna(stroke_data["bmi"].mean())
stroke_data.head()
np.shape(stroke_data)
(5110, 12)
stroke_data = stroke_data.drop("id", axis=1)
#label encoding
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
# gender male, female and other (1,0,2)
stroke_data["gender"] = labelencoder.fit_transform(stroke_data["gender"])
# work_type has Private = 2 'Self-employed = 3' ,Govt_job = 0', 'children = 4'
, 'Never_worked = 1 '
stroke_data["ever_married"] = labelencoder.fit_transform(stroke_data
["ever_married"])
# work_type has Private = 2 'Self-employed = 3' ,Govt_job = 0', 'children = 4'
, 'Never_worked = 1 '
stroke_data["work_type"] = labelencoder.fit_transform(stroke_data["work_type"])
# Residence_type 'Urban = 1 ' , 'Rural = 0'
stroke_data["Residence_type"] = labelencoder.fit_transform(stroke_data
["Residence_type"])
# smoking_status formerly smoked = 1', 'never smoked = 2', 'smokes = 3',
'Unknown = 0'
stroke_data["smoking_status"] = labelencoder.fit_transform(stroke_data
["smoking_status"])
data = stroke_data
data.head(1323)
# scaling the dataset using StandardScaler
from sklearn.preprocessing import StandardScaler

```

```

# scaling the dataset using StandardScaler
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
columns = ["age", "avg_glucose_level", "bmi"]
scaled = scaler.fit_transform(data[["age", "avg_glucose_level", "bmi"]])
scaled = pd.DataFrame(scaled, columns=columns)
data = data.drop(columns = columns, axis=1)
scaled.head(6)
data= pd.concat([data,scaled] ,axis=1)
data_df.head(700)
# separate the independent variables from dependent variables
X = data.drop(["stroke"],axis=1)
y = data["stroke"]
#prediction of stroke disease
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
# separate the independent variables from dependent variables
X = data.drop(["stroke"],axis=1)
y = data["stroke"]
# prediction of stroke disease
from collections import Counter
from imblearn.combine import SMOTEENN
#summarize class distribution
print("Before oversampling:", Counter(y))
#define oversampling strategie
smote = SMOTEENN(random_state=42)
# fit and apply the transform
x_smote,y_smote = smote.fit_resample(X,y)
#summarize class distribution

print("After oversampling:",Counter(y_smote))
#split train_test data
x_train,x_test,y_train,y_test = train_test_split(x_smote,y_smote,test_size=0.3,
random_state=42)
# scaling the dataset using StandardScaler
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
#ANOVA + logistic regression
from sklearn.feature_selection import SelectKBest, f_classif
from sklearn.pipeline import make_pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report

```

```

anova_filter = SelectKBest(f_classif, k="all")
logis = LogisticRegression(solver="lbfgs",max_iter=1000)
anova_logis = make_pipeline(anova_filter, logis)
anova_logis.fit(x_train, y_train)
pred_logis = anova_logis.predict(x_test)
print(classification_report(y_test, pred_logis))
confus = confusion_matrix(y_test, pred_logis)
sns.heatmap(confus, annot=True, cmap="Blues",linewidths=1,fmt="d")
plt.show()
#ANOVA + Algorithm
from sklearn.naive_bayes import GaussianNB
bayes = GaussianNB()
anova_filter_bayes = SelectKBest(f_classif,k='all')
anova_bayes = make_pipeline(anova_filter_bayes, bayes)
anova_bayes.fit(x_train, y_train)
pred_bayes = anova_bayes.predict(x_test)
print(classification_report(y_test, pred_bayes))
confus = confusion_matrix(y_test, pred_bayes)
sns.heatmap(confus, annot=True, cmap="Blues",linewidths=1,fmt="d")
plt.show()

```

```

from sklearn.neighbors import KNeighborsClassifier
anova_filter_KNN = SelectKBest(f_classif, k='all')
KNN = KNeighborsClassifier()
anova_KNN = make_pipeline(anova_filter_KNN, KNN)
anova_KNN.fit(x_train, y_train)
pred_KNN = anova_KNN.predict(x_test)
print(classification_report(y_test, pred_KNN))

sns.heatmap(confus, annot=True, cmap="Blues",linewidths=1,fmt="d")
confus = confusion_matrix(y_test, pred_KNN)
plt.show()

```

```

anova_filter_dss = SelectKBest(f_classif, k='all')
from sklearn.tree import DecisionTreeClassifier
dss =DecisionTreeClassifier()
anova_dss = make_pipeline(anova_filter_dss, dss)
anova_dss.fit(x_train, y_train)
pred_dss = anova_dss.predict(x_test)
print(classification_report(y_test, pred_dss))
confus = confusion_matrix(y_test, pred_dss)
sns.heatmap(confus, annot=True, cmap="Blues",linewidths=1,fmt="d")
plt.show()

```

#without ANOVA

```
from sklearn.linear_model import LogisticRegression
logis = LogisticRegression(solver="lbfgs",max_iter=1000)
logis.fit(x_train,y_train)
pred_logis = logis.predict(x_test)
print(classification_report(y_test,pred_logis))
confus = confusion_matrix(y_test, pred_logis)
sns.heatmap(confus, annot=True, cmap="Blues",linewidths=1,fmt="d")
plt.show()
```

```
bayes = GaussianNB()
from sklearn.naive_bayes import GaussianNB
bayes.fit(x_train,y_train)
pred_bayes = bayes.predict(x_test)
print(classification_report(y_test,pred_bayes))
confus = confusion_matrix(y_test, pred_bayes)
sns.heatmap(confus, annot=True, cmap="Blues",linewidths=1,fmt="d")
plt.show()
```

```
knn = KNeighborsClassifier()
from sklearn.neighbors import KNeighborsClassifier
knn.fit(x_train,y_train)
pred_knn = knn.predict(x_test)
print(classification_report(y_test,pred_knn))
confus = confusion_matrix(y_test, pred_knn)
sns.heatmap(confus, annot=True, cmap="Blues",linewidths=1,fmt="d")plt.show
()
```

References

- Abdelwahab B, Khalid N (2015) A model for predicting ischemic stroke using data mining algorithms. *Radiology* 275(2):510–520
- Aishwarya R, Navin K, Singh A, Kumar S (2018) Stroke prediction using decision trees in artificial intelligence. *Int J Adv Res Ideas Innov Technol* 4(2):1–7
- Alotaibi FS (2019) Implementation of machine learning model to predict heart failure disease. *Int J Adv Comput Sci Appl* 10(6):1–8
- Arslan AK, Colak C, Sarihan ME (2016) Different medical data mining approaches based prediction of ischemic stroke. *Comput Methods Prog Biomed* 130:87–92. <https://doi.org/10.1016/j.cmpb.2016.03.022>
- Bandi V, Bhattacharyya D, Midhunchakkravarthy D (2020) Prediction of brain stroke severity using machine learning. *Revue d'Intelligence Artificielle* 34(6):753–761. <https://doi.org/10.18280/ria.340609>
- Chiun-Li-Chin G-R, Wu B-J, Lin T-C, Weng C-S, Yang R-C, Su Y-JP (2017) An automated early ischemic stroke detection system using CNN deep learning algorithm. In: *IEEE 8th international conference on awareness science and technology*, pp 368–372

- Donkor ES (2018) Stroke in the 21st century: a snapshot of the burden, epidemiology, and quality of life. *Stroke Res Treat* 2018:3238165. <https://doi.org/10.1155/2018/3238165>. PMID: 30598741; PMCID: PMC6288566
- El Houari M (2019) Immediate applications of a new classification of finite dimensional simple lie algebras. In: *Nonassociative algebra and its applications*, pp 55–78. <https://doi.org/10.1201/9780429187674-6>
- Gangavarapu S, Gorli L, Aruna K (2021) Analyzing the performance of stroke prediction using ML classification algorithms. (IJACSA). *Int J Adv Comput Sci Appl* 12(6):539–544
- Javatpoint (2021) Confusion matrix in machine learning - Javatpoint. [Confusion Matrix in Machine Learning - Javatpoint](#)
- JoonNyung H, Jihoon G, Yoon H, Park Y, Dae K, Hyo S, Nam JH, Heo. (2019) Machine learning based model for prediction of outcomes in acute stroke. *Natl Libr Med* 50(5):1265–1265
- Kansadub T, Thammaboosadee S, Kiattisin S, Jalayondeja C (2015) Stroke risk prediction model based on demographic data. In: *8th biomedical engineering international conference (BMEiCON) IEEE*
- KDnuggets (2020) Decision tree algorithm. *Exp Dermatol*
- Liu YB, Cong Y (2017) The probability of ischaemic stroke prediction with neural-network model. *Front Imaging Neuro Sci* 9:233
- Luis Garcfa-Terriza RM (2019) Comparison of different machine learning approaches to model stroke subtype classification and risk prediction. *Society for Modeling & Simulation International (SCS)*
- Nwosu CS, Dev S, Bhardwaj P, Veeravalli B, John D (2019) Predicting stroke from electronic health records. In: *41st annual international conference of the IEEE engineering in medicine and biology society, IEEE*, pp 5704–5707
- Ohoud A (2018) Prediction of stroke using data mining classification techniques. *Int J Adv Comput Sci Appl* 9(1):475
- Pradeepa S, Manjula KR, Vimal S, Khan MS, Chilamkurti N, Luhach AK (2020) DRFS: detecting risk factor of stroke disease from social media using machine learning techniques. Springer, p 185
- Singh MS, Choudhary P, Thongam K (2020) A comparative analysis for various stroke prediction techniques. Springer, vol 11, 48, pp 98–106
- Sina Mirabdolbaghi SM, Amiri B (2022) Model optimization analysis of customer churn prediction using machine learning algorithms with focus on feature reductions. *Discret Dyn Nat Soc* 2022:1–20. <https://doi.org/10.1155/2022/5134356>