Stroke Prediction with Neural Networks
CMP6228

Jaspreet singh - ID:19150299

Bachelor of Science with Honours Computer and Data Science

School of Computing and Digital Technology

Faculty of Computing, Engineering and the Built Environment

Birmingham City University

Submitted June 2022

Contents

Abstract **1**

Introduction **1**

Problem statement **2**

Proposed method **3**

Experimental Results **5**

Summary **7**

Bibliography **7**

Abstract

A Stroke is a medical disorder in which regions of the brain are deprived of blood flow, resulting in a stroke, which can be deadly. Because the number of stroke cases is rising at an alarming rate, it is necessary to investigate the factors that influence the rate of growth of these instances. It is necessary to develop a method for predicting whether or not a person will be impacted by a stroke. This research examines machine learning techniques for better stroke prediction. With this in mind, this study employed deep learning techniques such as Neural Networks to create a model for an accurate prediction using a variety of physiological parameters. To detect the association between these distinct variables, various plots are used in the investigation. The research also demonstrates how certain characteristics, like as age, heart conditions, and smoking status, are essential, while others, such as residence, are not. As patients get older they are more at risk to suffer a stroke, heart problems greatly increase the risk of stroke, and the same is true for married people and those working in more stressful conditions. After the analysis and modeling, the algorithm has approximately reached an 80% accuracy rate with a recall of 0.6.

Introduction

"Every 4 minutes, someone dies of stroke." (Virani,[...], TT, 2020) As is known, cerebral stroke has become one of the main diseases endangering people's health and as stated by the World Health Organization (WHO) stroke is the second major cause of death around the world, responsible for approximately 11% of the total amount of deaths. By understanding the gravity of a stroke event and comprehending the risks of a stroke, mortality rates could decrease and more lives could be saved. An artifact can be produced and potentially be a useful aid for screening patients. It would be used in the process of determining whether an individual is at risk of stroke and allow medical intervention as a preventative matter if one is found to be at risk of stroke. So with the creation of a deep learning model to perform early detection of stroke for patients, it's possible to help assist the doctors in their job and with a higher prediction accuracy, the chances that this deep learning model will help prevent even more future occurrences of strokes are increased. Now to create this cited artifact, a neural network will be used to train and predict possible strokes in patients.

To complete this assignment, a dataset with diverse physiological features as attributes is taken from Kaggle. These characteristics are then examined and used to make the final forecast. The dataset is cleaned and prepared so that the machine learning model can interpret it. Thanks to Data Preprocessing, the dataset is examined for null values and filled accordingly with KNN imputer. Then, one hot encoding and label encoding is used to transform string values into integers followed by the use of the standard scaler or the min-max scaler. The dataset is then separated into train, test, and validation data then is balanced with SMOTE. The paper shows the implementation of Deep Learning classification algorithms by utilising a neural network, the model is then developed using this new data. The accuracy of the method is calculated and evaluated to determine the best tuning for the model's prediction performance. For this evaluation, Train and validation loss and accuracy graphs are plotted as well as a confusion matrix accompanied by the accuracy, recall, precision, and F1 score of the prediction. The model performed acceptably with the best parameters after tuning granting a good accuracy of around 75% - 80% with a discrete recall of 0.5 - 0.6.

Problem statement

Strokes deplete brain tissue of vital oxygen and nutrient-rich blood, causing brain cells to die. Every second is crucial. The sooner a stroke victim receives care, the greater his or her chances of recovery are. That is why it is critical to recognise the symptoms of a stroke as the longer the brain is deprived of oxygen, the more severe the outcome is. (NHS, 2019) Based on input criteria such as gender, age, different illnesses, and smoking status, this dataset is used to predict whether a patient is likely to have a stroke. Each row of data contains essential information about the patient and his or her behaviors. Every patient has different data that could have some correlation to their risk of having a stroke and these need to be detected first to create a better and more efficient deep learning model.

The dataset for stroke prediction is from Kaggle and with 5110 rows and 12 columns, this dataset is quite large. The main attributes of the columns are 'id', 'gender', 'age', 'hypertension', 'heart disease', 'ever married', 'work type', 'Residence type', 'avg glucose level', 'BMI', 'smoking_status', and 'stroke'. The value of the output column "stroke" is either a '1' or a '0.' The number '0' indicates that there is no risk of stroke, whereas the value '1' suggests that there is a chance of stroke. The potential of a '0' in the output column 'stroke' outnumbers the possibility of a '1' in the same column, resulting in a severely unbalanced dataset. Only 249 rows in the stroke column have a value of '1', whereas 4861 rows have a value of '0'. Data pre-processing is used to balance the data to improve accuracy. The dataset analysed above is shown in Table 1.

Table 1 - Stroke Dataset

Feature Name	Value Types	Description
1. id	Int	A unique id for patients
2. gender	String (Male, Female, Other)	Gender of the patient
3. age	Int	Age of the Patient
4. hypertension	Int (1, 0)	If the patient has hypertension
5. heart_disease Integer	Int (1, 0)	If the patient has heart disease

Feature Name	Value Types	Description
6. ever_married	String (Yes, No)	Patient's marital status
7. work_type	String (children, Govt_job, Never_worked, Private, Selfemployed)	work category of patients
8. Residence_type	String (Urban, Rural)	Patient's residence type
9. avg_glucose_level	Float	Average value of glucose level in patient's blood
10. bmi	Float	Patient's Body Mass Index
11. smoking_status	String (formerly smoked, never smoked, smokes, unknown)	Patient's smoking status
12. stroke	Int (1, 0)	Patient's stroke status (Output column)

Proposed method

Before developing a model, Data preprocessing is essential to eliminate undesirable noise and outliers from the dataset, which might cause a divergence from good training. This step takes care of everything that prevents the model from performing as efficiently as possible. Following the collection of the suitable dataset for Kaggle named 'Stroke Prediction Dataset', the data must be cleaned and checked to ensure that it is ready for model creation.

As seen in Table 1, the dataset is composed of 12 features. To begin, the column 'id' is removed because its presence in model construction makes no impact. After printing more information about the data frame and its unique values for each feature, is possible to drop the single line containing the only value 'Other' in the gender feature of the entire dataset as it would only increase the processing times and features after any possible encoding. The dataset is next searched for null values and, if any are discovered, they are filled. In this scenario, the null values in the column 'BMI' are filled using the KNN imputer with 6 'k' samples. These samples present in the dataset will be similar or close in the space and will help estimate the value of the missing data points by calculating the mean of the chosen samples. As the data contain categorical columns, these need to be encoded. In this artifact label encoder was adopted for the columns "gender", "ever_married" and "Residence_type" as this method will require less space and it's still applicable as all 3 columns contain only 2 different values. After this operation, all column labels are dropped and the encoding of the categorical features "smoking_status" and "work_type" using OneHotEncoder is executed. Now, before moving to the train_test_split the scaling of the numeric features "BMI", "age" and "avg_glucose_level" with StandardScaler or MinMaxScaler is done.

The dataset used for stroke prediction is extremely unbalanced. The total number of rows in the dataset is now 5109, with 249 rows indicating the likelihood of a stroke and 4860 rows indicating the absence of a stroke. Figure 1 shows a graphical illustration of the imbalance.

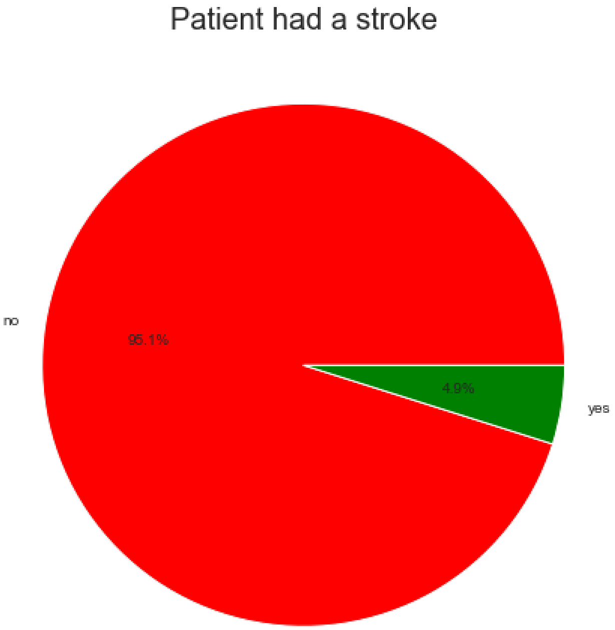


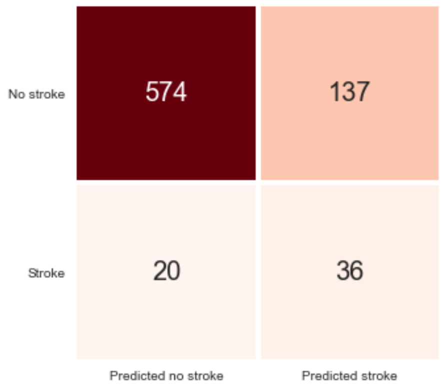
Figure 1 (pie chart of the imbalance between positive and negative likelihood of a stroke.)

The undersampled data is split into training and testing data for better accuracy and efficiency for this task keeping the ratio as 70% training data and 30% testing data which is then later split again with a ratio of 50/50 with the validation data.

While using such data to train a machine-level model may result in accuracy, other accuracy measures such as precision and recall are shallow. If such unbalanced data is not dealt with properly, the findings will be inaccurate, and the forecast would be ineffective. As a result, to develop an efficient model, this unbalanced data must be dealt with first. The method of oversampling is employed for this purpose by using SMOTE. Oversampling equalizes the data by oversampling the minority class to match the majority class. In this scenario, the class with the value "0" is oversampled in comparison to the class with the value "1." The training sample will be composed of 6832 rows that will include 3416 rows with value ’0’ and 3416 rows with value ’1’ after oversampling. After completing data preprocessing and handling the imbalanced dataset, as well as splitting the data, the next step is building the model.

Experimental Results

The model used is built in a specific callable function and this dense neural network is composed of 3 layers with the first having 64 units while the second 32 and the last 1. After the first 2 hidden layers, there is a dropout of 0.6 to disable neurons and prevent overfitting. These first 2 have also included the bias and kernel regularizer set at l2(0.01) to reduce overfitting again and have activation of “relu” that transforms the summed weighted input from the node into the activation of the node or output for that input. As for the last layer it only contains the activation as “sigmoid” as that guarantee that the output of this unit will always be between 0 and 1. When compiling the neural network it has an optimizer of “adam” and loss as binary_crossentropy as the target output should be 1 or 0. Finally, the learning rate is set to 0.0001 to have a much smoother line in the graph and decrease the potential loss of the model. Most of the parameters were suggested by the GridSearchCV after a long processing time, from the optimizer to the batch size of 64 and the epochs as 100. The Early stopping technique has also been used to facilitate the training as no matter the epoch chosen it will stop training once the model’s performance stops increasing on a validation dataset. Once done with the training with the model on the balanced train dataset, the history attribute will be used to plot the recorded training and validation loss and metrics values at successive epochs. Other evaluation information is also printed in the console to have a better understanding of the results such as the min value for loss, and max value for accuracy for both the validation and training set. As for the test evaluation the accuracy, recall, precision, and F1 score are printed, lastly, a confusion matrix with the predictions of the test has been also plotted. All this information is shown in Figures 2 and 3 below.



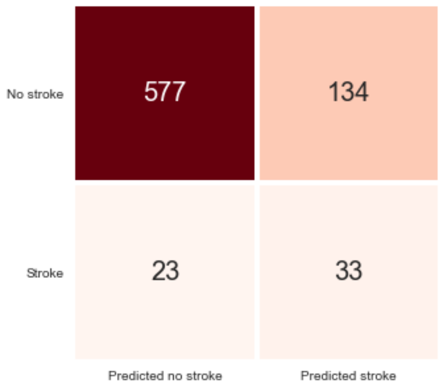
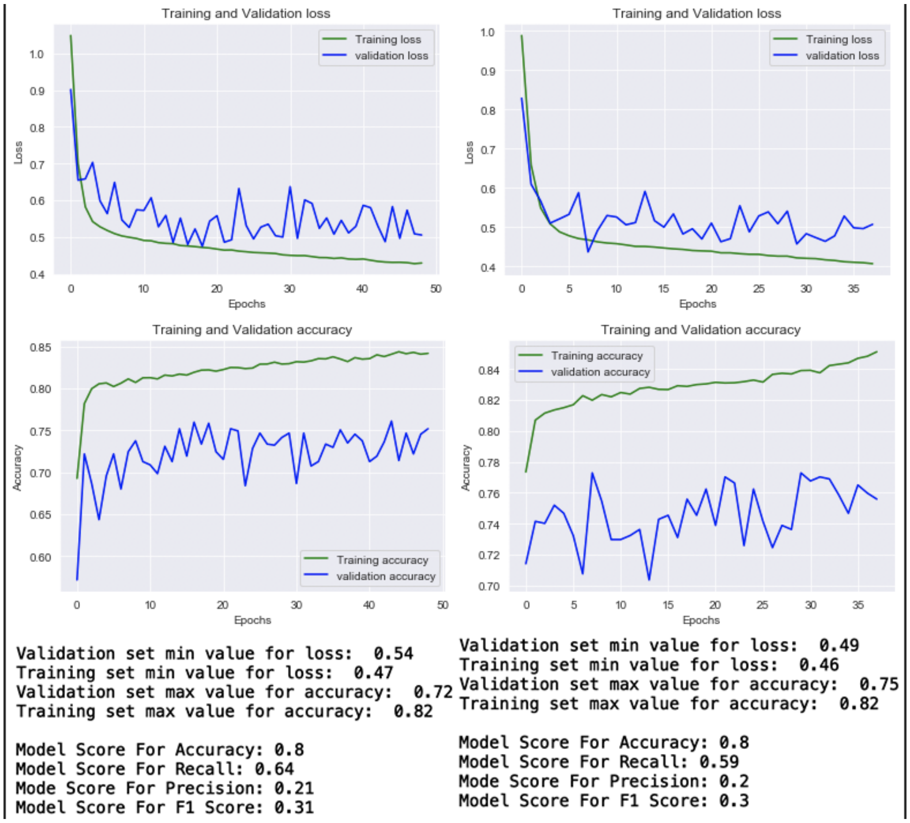


Figure 2 (Plot of confusion matrix. On the right using standard scaler and minmax scaler on the left.)



```
Validation set min value for loss:  0.54
Training set min value for loss:  0.47
Validation set max value for accuracy:  0.72
Training set max value for accuracy:  0.82

Model Score For Accuracy: 0.8
Model Score For Recall: 0.64
Mode Score For Precision: 0.21
Model Score For F1 Score: 0.31
```

```
Validation set min value for loss:  0.49
Training set min value for loss:  0.46
Validation set max value for accuracy:  0.75
Training set max value for accuracy:  0.82

Model Score For Accuracy: 0.8
Model Score For Recall: 0.59
Mode Score For Precision: 0.2
Model Score For F1 Score: 0.3
```

Figure 3 (Plot of train and validation loss and accuracy and other evaluation scores while using both scalers. On the right by using standard scaler and on the left using minmax scaler.)

While evaluating the results, it’s clear that the validation loss and accuracy are not as smooth as the training, this could be caused by the fact that the validation set used is small compared to the training set thus making bigger steps even after decreasing the learning rate. Looking at the loss graphs, especially for the standard scaler is possible to see some sign of overfitting as the loss is increasing near the last plotted epochs. A possible solution to these points could be dropping the features with less correlation. In retrospect, the possibility to use any of both scalers doesn’t seem to create much difference in the results as in both cases the scores for the prediction are quite similar and both seem to have acceptable results for accuracy and recall while having a low precision and F1 score. A final note is the implementation of a threshold of 0.2 to induce the model of classifying the prediction as a “stroke” rather than a “no stroke” as in this situation it is better to have a higher false-positive rate than a false negative rate this means that around 18% of patients are being falsely identified as at risk.

Summary

Stroke is a serious medical illness that must be addressed as soon as possible. The development of a machine learning model can aid in the early detection of stroke and lessen the severity of future consequences. The efficacy of several machine learning algorithms in accurately predicting stroke based on multiple physiological parameters is demonstrated in this research. With an accuracy of 75 to 80 % and a recall of 0.5 to 0.6, the neural network Classification outperforms all its previous model versions during the creation of this artifact. The study also shows how some criteria are critical while some others are not, this could indicate that a possible future development could focus more on the feature selection and drop columns that have shown little to no correlation to the risk of a stroke. Another note could be to increase the recall despite lowering the accuracy as in the case of the study it's important to predict positive cases as much as possible.

Bibliography

www.kaggle.com. (n.d.). Stroke Prediction Dataset. [online] Available at: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset [Accessed 11 Mar. 2022].

Virani, S.S., [. . .], T.T. (2020). Heart Disease and Stroke statistics—2020 Update. Circulation, [online] 141(9). Available at: https://www.ahajournals.org/doi/10.1161/CIR.0000000000000757.

NHS (2019). Stroke. [online] NHS. Available at: https://www.nhs.uk/conditions/stroke/.