Stroke prediction using SVM, kNN, and SNN machine learning algorithms

S. Fraile - 2659410 - s.m.fraile@vu.nl

Vrije Universiteit Amsterdam

26-03-2021

Abstract. This paper discusses the research question of whether one of the proposed machine learning methods can improve the performance of predicting strokes based on biometric and lifestyle data. This research question was tested using three different machine learning algorithms, such as the K-Nearest-Neighbours (KNN), the Support Vector Machine (SVM) and finally the Simple Neural Networks (SNN). Our results consisted of measuring the performance of the three algorithms using accuracy, precision/recall and F1 score. Finally, based on their measured performance, it could be determined which of the three algorithms performs best when predicting stokes based on lifestyle and biometric data.

Keywords: Stroke Prediction · Support Vector Machines (SVM) · K-nearest neighbours (KNN) · Neural Networks · Synthetic Minority Over-sampling Technique (SMOTE) · Edited Nearest Neighbor (ENN) · SMOTEENN · Electronic medical claims (EMCs) · Body mass index (BMI) · Performance · Accuracy · Precision/Recall · F1

1. Introduction

1.1. Background

Cerebrovascular accidents, commonly known as strokes, are not only the third leading cause of disabilities, but they are also the second leading cause of death on a worldwide scale [1] and projections indicate that it will remain so in the year 2020 [14], furthermore, it creates a big impact economically as 10%-15% of all stroke patients are young adults that are left victims to disability before their productive years [14]. A stroke is classified as a disease that is caused by bleeding, or a rupture or blockage in the blood vessel or artery that delivers nutrients and oxygen to the brain. Even though medical advancements have tremendously aided in recovering various patients from strokes, several common underlying risk factors that are still prevalent in the lifestyles of many individuals are not as easily predictive and preventable of a stroke. Such lifestyle risk factors include an unhealthy diet, smoking, diabetes, and much more. It is, therefore, crucial to target such individuals who are at high risk of a hemorrhagic or ischemic stroke and have preventive measures to reduce one of the top worldwide leading causes of death.

One important aspect to note about the importance of prevention is that treatment post-facto is costly, leaving patients in intensive care units for long periods and it is not always successful at allowing the individual to reintegrate into normal activities. [15]

Large international organizations, such as the World Stroke Organization, World Health Organization (WHO), and the World Federation of Neurology, have taken initiatives to raise awareness and advocate better data collection to more successfully predict and prevent strokes [1]. Such initiatives are crucial for data-driven predictive algorithms. This paper focalizes on different Machine Learning algorithms which predict stroke occurrence promptly to aid early medical intervention.

1.2. Related work

There have been several other research papers investigating ML algorithms that predict stroke occurrences. Chen-Ying, et.al compared in the 2017 study several algorithms including deep neural networks in stroke prediction. Deep neural networks were selected in this paper for their impressive results in other areas such as speech recognition, and this paper shows DNN as the highest rate of success in the prediction of strokes [16]. Another paper also references DNN as an approach to predicting post-stroke consequences such as pneumonia [17].

The use of machine learning techniques and algorithms in the hope of creating prevention for certain medical conditions is in itself a great area of opportunity. Tianyu Liu, *et. al* dives also into the prediction of strokes with a hybrid approach using random forest regression for missing values and an automated hyperparameter optimization based on deep neural networks, with effective results in reducing false negatives in the data set [7].

Deep neural networks are the preferred algorithm for most of the related work in the prediction of strokes [6]. With that in mind, and with the idea of not being redundant in our research we have decided to investigate different algorithms to test their accuracy in the prediction of strokes.

1.3. Research question

This paper explores several machine learning algorithms including K-Nearest-Neighbours (KNN), the Support Vector Machine (SVM) and the Simple Neural Networks (SNN) with backpropagation, which all will be trained to aim to construct a model that predicts strokes. These algorithms will be compared by several metrics discussed further throughout the paper to determine which model offers the most optimal classifier for stroke prediction given the data. The research question for this paper is: "*Which of the three machine learning algorithms introduced above offers a more optimal model for the prediction of strokes based on lifestyle and biometric data*?"

2. Data Analysis

2.1. Data inspection

All models discussed in this paper have been trained on a stroke prediction online dataset from the dataset library Kaggle [2]. This dataset contains 5110 instances with several input parameters, which are the features of the dataset that include biometric and lifestyle data. The biometric data consists of the patient's ID, gender, age, whether a person has experienced hypertension and has had heart diseases before, glucose levels, BMI. Lifestyle data includes marital status, work status, residence type, and smoking habits. The output or target value is whether the patient had any occurrence of a stroke.

Upon inspection, several instances had extreme BMI values, these were considered outliers (see Figure 1). Nonetheless, it is essential to take into account individuals with more extreme weight because weight can be seen as a sensitive attribute in the real world as there are numerous individuals with varying BMIs.

Figure 1. Boxplot showing data distribution and outliers

In addition, there is one instance that has 'Other' as the gender value. In this particular case, it is important to annotate this instance since gender is one of the most important sensitive attributes and this could help us to study bias. Therefore, it is important to include this instance to be all-encompassing of non-binary individuals as well as this promotes more socially aware models.

Moreover, during the inspection of the dataset, it was noticed that 201 instances contain N/A values in the BMI feature, which were undefined, and more than 1000 instances contained "Unknown" in the smoking status feature. These attributes have a high chance of causing overfitting. Since the total number of instances is

5110, it would not be wise to remove the ones with undefined or unknown values. Therefore, to resolve the issue of having missing data, data imputation has to be performed.

Lastly, a recurring problematic aspect for many of the studies is that most medical datasets have a natural class imbalance. Class imbalance refers to the issue of having the number of instances in each class in the training set heavily skewed towards the majority class, in this context the minority class is the stroke occurrence class. As can be seen in figure 2, there is a heavy class imbalance in this dataset as well. This class imbalance is within the target value of having a stroke, where there are only 249 instances that have a target value of having a stroke compared to 4861 instances not having a stroke. This happens to be a critical issue for disease prediction algorithms as a class imbalance may cause the classifier to be biased towards the majority class, meaning that it may cause less accurate and valid results for predicting the disease, which impairs the objective of such algorithms in the medical field.

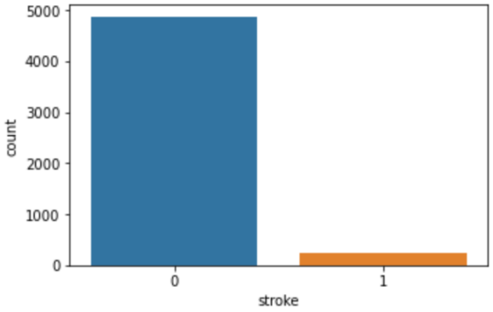


Figure 2. Class imbalance in stroke target values before upsampling

2.2. Data Preprocessing

During this research, three machine learning algorithms have been used to predict stroke based on the given data. To be able to perform the stroke prediction, the data had to be preprocessed accordingly to be in an appropriate format. This format included the data being split into a training and test set, which included 75% of the data being used for the training of the models and 25% being used for the testing. Besides, by using the Cross-validation method, the optimal hyperparameters were searched on the validation data without the fear of multiple testing.

More specifically, a 5-fold Cross-Validation was performed in our models to choose the hyperparameters. For this task, the training set was randomly split into 5 parts, each consisting of training and test data. Then, during 125 iterations, a subset was left out, this subset became the Validation set, and a model was built on the remaining 4 sets. The validation data was used to optimize hyperparameters for the proposed algorithms. Furthermore, the hyperparameters were selected by averaging the scores of the iterations.

Secondly, one of the feature columns has been removed, which consisted of the patients' unique id numbers. These numerical values do not influence the proposed research question and therefore were irrelevant for the models of stroke predictions.

Individuals leaving or not responding to surveys often results in data collection failures. Inaccurate predictive models are often the product of missing data. This missing data can be restored using data imputation. Therefore, to tackle the issue of having unknown and undefined values in our features that are described in Section 2.1, data imputation is necessary. These values were imputed by using the mean or the mode values. More precisely, for the BMI values, the average values across the column were used to impute the N/A values. Using the mean is described to be the most common method for data imputation with numerical variables [9, 10]. Whereas, for the smoking status values, which are categorical, the mode imputation method was used to impute the unknown values [10].

Additionally, the One-hot-Encoder method has been utilized to encode categorical features as a one-hot numeric array. Therefore, lifestyle features such as marital status, work status, residence type, and smoking habits were represented as binary vectors.

Lastly, researchers found several methods to counteract and prevent the effects of having a class imbalance, where of these methods includes upsampling with the SMOTE method [8]. This algorithm generates new synthetic observations that do not yet exist in the data from the minority class instances and their randomly chosen nearest neighbours, which in our case are the instances classified as having a stroke. It then adds these new samples to the dataset and uses that to train the models. By using this technique of upsampling and thus creating new observations from instances of having a stroke, we increase the size of the minority class, which results in a more balanced dataset, simulates variation in the data and may reduce the chance of overfitting [13], but does not eliminate it entirely. The correct way of handling imbalanced data is to perform this SMOTE upsampling method during cross-validation, thus for each fold, where upsampling is performed before training and this process is repeated for each fold [13].

3. Models

In this section, the three machine learning models will be discussed. These models are the K-Nearest-Neighbours method, the Support Vector Machine method and the Neural Network method.

3.1. K-Nearest-Neighbours

The K-Nearest-Neighbours method (KNN) uses non-generalizing learning; this means that instead of attempting to create a general internal model during learning, the KNN model stores the instances of the training data. When using the KNN algorithm, classification is computed by calculating for each class the representatives within the nearest neighbours of each point, and subsequently classifying it by the majority vote of the nearest neighbours. The number of neighbours, 'k', is a hyperparameter which is an integer-value that is specified by the user. The optimal value of the value k depends on the structure of the data. [3]

3.2. Support Vector Machine classification

The Support Vector Machine method (SVM) is a popular supervised learning method that can be used for either classification, regression or outlier detection. Support Vectors machines are known to be effective in high dimensional spaces due to their versatility. The versatility of the SVM can be ascribed to the kernel function which can be specified by the user. Kernel functions can either be linear, polynomial, RBF or sigmoid functions. A potential downside of using an SVM classifier can be it's susceptibility to overfitting when the number of features succeeds the number of samples in the data, which should be avoided by using the right kernel hyperparameters and possibly additional regularization [4].

3.3. Simple Neural Network

The Simple Neural Network method (SNN) used here is the Multi-layer Perceptron Classifier (MLP), which is a supervised learning algorithm that can learn a function f: $R^m \rightarrow R^o$ through training on a dataset. Here, m, is the dimensions of the input and o is the number of dimensions of the output. The MLP can learn a nonlinear function approximator for both regression and classification purposes when a set of features and a target are given. Between the input and output layers, there can be multiple nonlinear layers, also called hidden layers. The Multi-Layer Perceptron uses backpropagation to adjust the weights of the neurons accordingly. The advantage of using an MLP is their capability of learning nonlinear models, while disadvantages can be that the used loss function is non-convex leading to multiple local minima, and the usage of a large number of hyperparameters such as the number of hidden neurons. An additional drawback is that the layers and iterations have to be specified by the user. [5]

4. Method

Here, we still need to discuss grid search and performance metrics.

4.1. Grid Search as a hyperparameter search

For each of the implemented models it holds that there are parameters which are not directly learned and thus have to be specified by the user themself. This hyperparameter-space has to be searched in order to find an optimal validation score. Grid search is considered to be one of the formal methods of searching the hyperparameters for the proposed models. This method defines a finite set of values per hyperparameter and tries all combinations exhaustively. All combinations are subsequently evaluated using cross-validation, retaining the best combination of parameters. [18]

4.2. Performance Metrics

4.2.1. Accuracy

4.2.2. Precision

4.2.3. Recall

4.2.4. F1 score

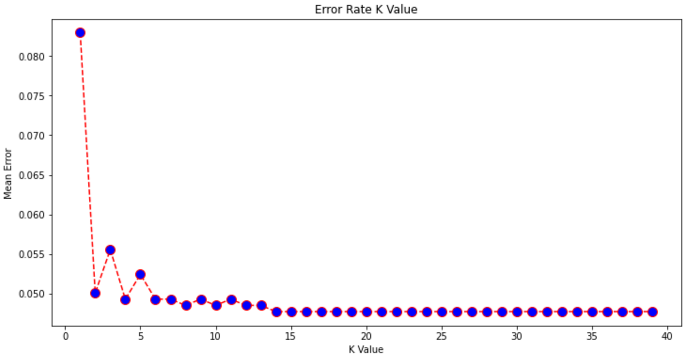
5. Results

All three machine learning algorithms discussed in Section 3, have been trained and tested in Python. The testing of the models has been done of 25% of the data, as mentioned before in Section 2.2. The performance of each model was examined based on three different metrics, which are accuracy, precision/recall and the F1-score metric. In the following subsections, the three machine learning models are displayed together with their results by illustrating each of the three performance metrics.

5.1. K-Nearest-Neighbours

The kNN algorithm was cross-validated using 12 different K values. In figure 3, different K-values were tested ranging from 1 to 40. There is a relatively simple trend, as the k-value increases the mean error decreases until an increase in K has a negligible effect. The optimum value for K, the best estimator, was identified as being 2. Further increases in the value for K make such little difference that it is not reasonable to increase it. The accuracy of the algorithm was found to be ___, the precision was found to be ___ and the F1 score was _____. The differences between these values indicate _______.

Figure 3.



5.2. Support Vector Machine

5.3. Simple Neural Network

At each step of our simple neural network, the multi-layer perceptron classifier calculates the partial derivative of the loss function per the model parameters to update the parameters resulting in an iterative training method. To maximize the effectiveness of our simple neural network, we performed grid-search with cross-validation using 3 splits. We had initial predictions for which parameters would be best such as the 'ADAM' solver which works best on large datasets of more than 1000 instances [12] but found that the SGD (stochastic gradient descent) solver produced better results.

6. Discussion

By comparing the accuracy, the precision/recall and the F1-score metrics of the models we can determine which of the three models offer an optimal model for predicting strokes based on biometric and lifestyle data. An overview of the applied machine learning methods and their performance can be seen in Table 1 below.

Model	Accuracy	Precision	Recall	F1-score
KNN				
SVM				
SNN with backpropagation				

Table 1: Performance for each of the three models

7. Conclusion

We created three models to investigate our research question of "Which machine learning algorithm of SVM, kNN, and SNN with back-propagation offers an optimal model for predicting strokes based on biometric and lifestyle data?". Based on the results obtained and the performance measured in sections 4 and 5, we can conclude that …

Given the complex nature of stroke inducing factors, further improvements can explore the DNN algorithm rather than SNN, since the DNN contains more layers that can be more representative of the nature of stroke occurrences [6].

Additionally, several undersampling and other new upsampling methods may be used in the future study to determine if they affect the complexity of the initial datasets. Finally, future studies should consider extending this analysis to include datasets with higher dimensionality, with a dataset containing a larger number of instances.

8. References

[1] Johnson, W., Onuma, O., Owolabi, M., & Sachdev, S. (2016, September). Stroke: a global response is needed. *Bulletin of the World Health Organization*, *Vol. 94*. http://dx.doi.org/10.2471/BLT.16.181636

[2] Fedesoriano (2021, January 26). Stroke Prediction Dataset, Version 1. Retrieved from: https://www.kaggle.com/fedesoriano/stroke-prediction-dataset

[3] Scikit-learn Developers (2010). 1.6.2. Nearest Neighbors Classification — documentation. Scikit-learn.org. Available: https://scikit-learn.org/stable/modules/neighbors.html#classification

[4] Scikit-learn Developers (2010). 1.4. Support Vector Machines — documentation. Scikit-learn.org. Available: https://scikit-learn.org/stable/modules/svm.html#svm

[5] Scikit-learn Developers (2010). 1.17. Neural network models (supervised) — scikit-learn 0.24.1 documentation. Scikit-learn.org. Available: https://scikit-learn.org/stable/modules/neural_networks_supervised.html

[6] Heo, J., Yoon, J. G., Park, H., Kim, Y. D., Nam, H. S., & Heo, J. H. (2019). Machine Learning-Based Model for Prediction of Outcomes in Acute Stroke. *Stroke*, *50*(5), pp. 1263–1265. https://doi.org/10.1161/strokeaha.118.024293

[7] Liu, T., Fan, W., & Wu, C. (2019, October 23). A hybrid machine learning approach to cerebral stroke prediction based on imbalanced medical dataset. *Artificial Intelligence in Medicine*, *Vol. 101*. https://doi.org/10.1016/j.artmed.2019.101723

[8] Chawla, N. V., Bowyer, K.W., Hall, L. O., & Kegelmeyer, W. P. (2002, June 2). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *Vol. 16*, pp. 321–357. https://doi.org/10.1613/jair.953

[9] Amballa, A. (2017, August 17). Feature Engineering Part-1 Mean/ Median Imputation. *Analytics Vidhya*. Retrieved from: https://medium.com/analytics-vidhya/feature-engineering-part-1-mean-median-imputation-761043b95379

[10] Van der Meijs, A. (2018, July). Missing Data Imputation: Predicting Missing Values. (Data Science). Retrieved from: http://arno.uvt.nl/show.cgi?fid=146868

[11] Idakwo, G., Thangapandian, S., Luttrell, J. *et al.* (2020, October 27). Structure-activity relationship-based chemical classification of highly imbalanced Tox21 datasets. *Journal of Cheminformatics 12, Vol.* 66. https://doi.org/10.1186/s13321-020-00468-x

[12] Scikit-learn Developers (2010). sklearn.neural_network.MLPClassifier — scikit-learn 0.20.3 documentation. Scikit-learn.org. Available: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

[13] Santos, M. S., Soares, J. P., Araujo, H., Santos, J., & Abreu, P. (2018, October). Cross-Validation for Imbalanced Datasets: Avoiding Overoptimistic and Overfitting Approaches. *IEEE Computational Intelligence Magazine*, *Vol. 13*, pp. 59–76. DOI: 10.1109/MCI.2018.2866730

[14] Smajlović, D. (2015). Strokes in young adults: epidemiology and prevention. *Vascular health and risk management*, *Vol. 11*, pp. 157–64. DOI: 10.2147/VHRM.S53203

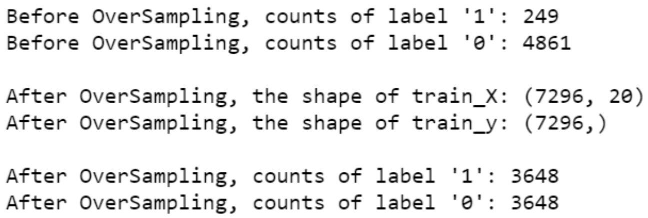
[15] Brott, T., & Bogousslavsky, J. (2000, September). Treatment of Acute Ischemic Stroke. *New England Journal of Medicine*, *Vol. 343*, *No. 10*, pp. 710–722. DOI: 10.1056/nejm200009073431007

[16] Hung, C., Chen, W., Lai, P., Lin, C., & Lee, C. (2017, July). Comparing deep neural network and other machine learning algorithms for stroke prediction in a large-scale population-based electronic medical claims database. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 3110-3113. DOI: 10.1109/EMBC.2017.8037515

[17] Ge, Y., Wang, Q. *et al.* (2019, December). Predicting Post-stroke Pneumonia Using Deep Neural Network Approaches. *International Journal of Medical Informatics*, Vol. 132. DOI: 10.1016/j.ijmedinf.2019.103986

[18] Scikit-learn Developers (2021). Exhaustive Grid Search — scikit-learn 0.24.1 documentation. Scikit-learn.org. Available: https://scikit-learn.org/stable/modules/grid_search.html#grid-search

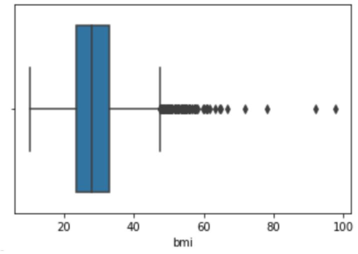
A. Appendix



```
Before OverSampling, counts of label '1': 249
Before OverSampling, counts of label '0': 4861

After OverSampling, the shape of train_X: (7296, 20)
After OverSampling, the shape of train_y: (7296,)

After OverSampling, counts of label '1': 3648
After OverSampling, counts of label '0': 3648
```


bmi