Impact of Class Weights and Feature Importance in Automated Stroke Detection

Final Paper

Avyukth Harish

11/16/2022

Inspirit AI Research Project

Abstract

A stroke occurs when a blood vessel that carries oxygen and nutrients to the brain is either blocked by a clot or ruptures. Consequently, part of the brain is unable to obtain the blood (and oxygen) it needs, so brain cells die. This makes it important to be able to assess the probability of a stroke given features that are specific to patients so that they could take preventative measures in the future. Thus, the unpredictability and deadliness of strokes pose the following questions: Can we predict the occurrence of a stroke from few facts about the patient easily accessible by the doctor? What are the most important features for stroke prediction? In this project, we investigate the feasibility of using a supervised machine learning model to predict stroke occurrence. In practice, however, we faced challenges such as low prevalence and the imbalance in the available dataset, with many more negative than positive cases. In this research paper, we do a parametric study of class weighting as a way to tackle imbalance during training. We then infer the most important features that should be taken into consideration for stroke prediction. Assessing feature importance allows for patients to focus on two or three areas that may be contributing to their high probability for getting a stroke. The most significant result was that the most important feature that should be considered when determining the probability of an individual getting a stroke is age. However, there is no distinct second most important feature. Additionally, a non monotonic improvement with a class weight of 22.5 for positive cases in this dataset produces the most optimal results.

1. Introduction

There are several factors that can play a role in the chances of getting a stroke. As outlined in the McKinsey Analytics Online Hackathon Stroke Detection Dataset, some of these factors include gender, age, hypertension, glucose level, BMI, and smoking status. Symptoms of strokes include trouble walking, speaking, and understanding, as well as paralysis or numbness of the face, arm, or leg. The severity and unpredictability of strokes highlight the importance of prediction and preventative measures. Thus, automated methodology is crucial in this field as it provides a method for early detection. There are several challenges faced in the process of creating a suitable model for stroke detection, one of which includes the severe imbalance present in the dataset with a larger number of negative test cases. In order to address this problem, we create class weights to balance the dataset and produce balanced results based on a variety of accuracy metrics. Additionally, to address the question of feature importance, we use mean decrease in impurity to determine which feature is the most significant in assessing the probability of an individual getting a stroke. In this research paper, we first discuss the dataset itself and preprocessing and encoding techniques utilized to create a favorable format of data to input into our model. We then discuss the methodology and models implemented, and the results that follow.

2. Background

Numerous studies centered around stroke detection have been recently conducted, two of which include "Identifying Stroke Indicators Using Rough Sets" by Pathan et al. (2020) [2] and "A predictive analytics approach for stroke prediction using machine learning and neural networks" by Dev et al. (2022) [3]. The former article focuses on rough sets in the data preprocessing stage through methods such as down-sampling of the dataset. A potential problem with this could be the fact that several good data points resulting in negative test cases are thrown away in an effort to balance the number of positive and negative test cases. Thus, an alternative method could be to assign weights to positive cases in order to balance the dataset in a manner which keeps all the relevant data while creating a balanced dataset, a key focus in this research project. The latter article focuses on creating the highest accuracy model with multi-layer perceptrons such as convolutional neural networks and neural networks in addition to balancing the dataset using sub-sampling techniques. While such models could potentially have higher accuracies, models such as CNNs are typically more suited for images rather than tabular data. In this sense, an alternative approach could be to use a classification Random Forest model.

In this study, we aim to gain a thorough understanding of the impact of the weights assigned to each sample in the loss function used to optimize the model parameters. To this effect, we employ a tree-based model, Random Forest, and conduct a parametric study on the class weight assigned to positive samples. Contrary to other studies seeking state of the art accuracy, we instead focus on understanding the role played by critical parameters.

3. Dataset

The dataset used in this project was the McKinsey Analytics Online Hackathon Stroke Detection Dataset [1]. This dataset consists of tabular data with several possible features affecting the likelihood of a patient having a stroke, in addition to binary labels of either '1' or '0' to indicate whether or not the patient did have a stroke. A total of 5110 data points were used, and each data point or patient consisted of 10 total features. The 3 numerical features include age, BMI, and average glucose level while the 7 categorical features include gender, hypertension, heart disease, ever married, work type, residence type, and smoking status.

Data preprocessing is required to encode all the features in order to implement a Random Forest model. In order to encode the categorical variables, we utilize the get_dummies function as a part of pandas, creating a usable format for the Random Forest Classifier. Two critical parameters include dummy_na and drop_first. We set dummy_na as false for all categorical variables since we verified during the data exploration phase that there were no missing values for any of the categorical variables. Additionally, we set drop_first as true for all binary variables but false for all variables with more than 2 categories to have a balanced representation of all possible feature

values. Since we have missing values in one of the columns, BMI, we remove all the missing values. After preprocessing, we maintain a total of 16 features.

For the validation, we used a 90-10 train test split. We use stratified splitting to ensure that the validation sets have the same imbalance properties for uniform results. We also utilize cross-validation to get more robust estimates of our evaluation metrics.

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1
...	...	...	...	...	...	...	...	...	...	...	...	...
5105	18234	Female	80.0	1	0	Yes	Private	Urban	83.75	NaN	never smoked	0
5106	44873	Female	81.0	0	0	Yes	Self-employed	Urban	125.20	40.0	never smoked	0
5107	19723	Female	35.0	0	0	Yes	Self-employed	Rural	82.99	30.6	never smoked	0
5108	37544	Male	51.0	0	0	Yes	Private	Rural	166.29	25.6	formerly smoked	0
5109	44679	Female	44.0	0	0	Yes	Govt_job	Urban	85.28	26.2	Unknown	0

5110 rows × 12 columns

Table 1: Initial tabular dataset with numerical and categorical variables.

	0	1	2	3	4	5	6	7	8	9	...	5100	5101	5102	5103	5104	5105	5106	5107	5108	5109
age	67.00	61.00	80.00	49.00	79.00	81.00	74.00	69.00	59.00	78.00	...	82.00	45.00	57.00	18.00	13.00	80.00	81.0	35.00	51.00	44.00
avg_glucose_level	228.69	202.21	105.92	171.23	174.12	186.21	70.09	94.39	76.15	58.57	...	71.97	97.95	77.93	82.85	103.08	83.75	125.2	82.99	166.29	85.28
bmi	36.60	NaN	32.50	34.40	24.00	29.00	27.40	22.80	NaN	24.20	...	28.30	24.50	21.70	46.90	18.60	NaN	40.0	30.60	25.60	26.20
stroke	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.0	0.00	0.00	0.00
gender_Male	1.00	0.00	1.00	0.00	0.00	1.00	1.00	0.00	0.00	0.00	...	1.00	0.00	0.00	0.00	0.00	0.00	0.0	0.00	1.00	0.00
hypertension_1	0.00	0.00	0.00	0.00	1.00	0.00	1.00	0.00	0.00	0.00	...	1.00	0.00	0.00	0.00	0.00	1.00	0.0	0.00	0.00	0.00
heart_disease_1	1.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.0	0.00	0.00	0.00
ever_married_Yes	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00	1.00	1.00	...	1.00	1.00	1.00	0.00	0.00	1.00	1.0	1.00	1.00	1.00
Residence_type_Urban	1.00	0.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	...	0.00	1.00	0.00	1.00	0.00	1.00	1.0	0.00	0.00	1.00
work_type_Never_worked	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.0	0.00	0.00	0.00
work_type_Private	1.00	0.00	1.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00	...	0.00	1.00	1.00	1.00	0.00	1.00	0.0	0.00	1.00	0.00
work_type_Self-employed	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	...	1.00	0.00	0.00	0.00	0.00	0.00	1.0	1.00	0.00	0.00
work_type_children	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	1.00	0.00	0.0	0.00	0.00	0.00
smoking_status_formerly smoked	1.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.0	0.00	1.00	0.00
smoking_status_never smoked	0.00	1.00	1.00	0.00	1.00	0.00	1.00	1.00	0.00	0.00	...	1.00	0.00	1.00	0.00	0.00	1.00	1.0	1.00	0.00	0.00
smoking_status_smokes	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.0	0.00	0.00	0.00

16 rows × 5109 columns

Table 2: Preprocessed sample of dataset, the table is transposed for visual clarity.

4. Methodology / Models

For the modeling portion of the research, we start by splitting the dataset into validation sets, with a 90-10 split using stratified cross validation. Then, we create our Random Forest model with default values from scikit-learn except for a max depth of 5 to avoid overfitting. After creating this model, we tackle the problem of creating weights. Due to a severe imbalance in the dataset (more negative than positive cases), we take a systematic approach to determining the relationship that different positive weights had towards different accuracy scores. We use the loss function Binary Crossentropy as our end goal for our model is binary classification. This in fact uses weights, and the focus of this study is to understand the role that the weights play. The model parameters are trained by optimizing the loss function. Thus, the choice of the loss function affects the choice of the parameters therein.

In order to determine the proper weight for positive cases, we create an alpha_range, an array storing powers of 2 from -2 to 10. We then loop through the values in alpha_range and assign each respective alpha to positive samples while keeping negative samples with a weight of 1. In doing so, we assign this class weight to the class weight parameter in the Random Forest model and based our results on the following 3 accuracy metrics: balanced accuracy, recall, and precision. These two accuracy metrics are absolutely critical because of the imbalance. However, the tradeoff between the two makes it hard to determine a specific alpha value. For example, a model that predicts every single patient as likely to get a stroke will attain perfect recall but poor precision. At the same time, a model that predicts very few as likely to get a stroke can have better precision but poor recall. Thus, it is important to take the clinical implications into consideration when determining the right alpha value. For each of these values, we separately evaluate performance metrics using stratified cross validation.

After determining the proper weight for positive samples and the subsequent accuracy scores, we calculate the importance of each feature for stroke prediction through the use of the mean decrease in impurity.

5. Results and Discussion

After utilizing the range of alpha values in order to compare the different accuracy metrics such as balanced accuracy score, precision, and recall we determine the optimal alpha value to be 22.5, producing a balanced accuracy score, recall, and precision score of 71%, 65%, and 11% respectively. We achieve this by utilizing the ‘class_weight’ parameter in the Random Forest Classifier model. While there is no way to precisely determine which alpha value should be used, we approximate this value based on the importance of precision and recall in this scenario, and the priority of correctly diagnosing someone who is likely to get a stroke above all else. However, while keeping this in mind, it is still important to have some sort of balance between the three metrics, demonstrating why the chosen alpha value is not the highest possible value. The alpha value of 22.5 shown in relation to the three accuracy metrics can be seen below.

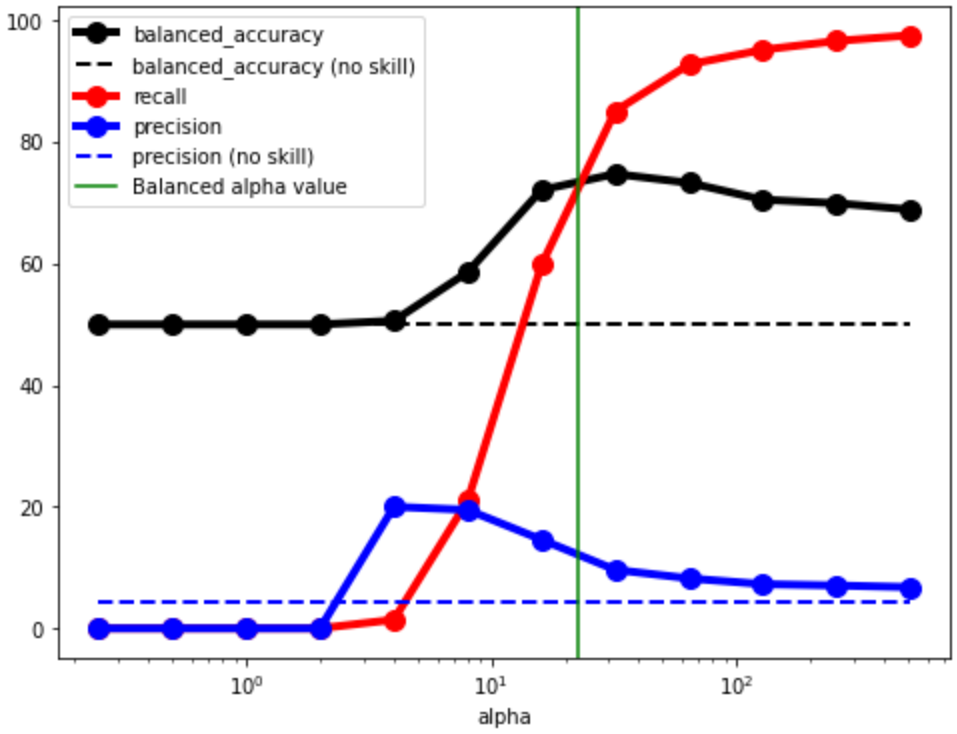

Figure 1: A precision, recall, balanced accuracy, and alpha graph displaying the relationship between a set of alpha values and the aforementioned accuracy metrics. An alpha value of 22.5 is shown in relation to the three accuracy metrics to present their respective accuracy scores.

As shown above, we create a baseline accuracy for the model, one that was determined by predicting the accuracy if the model only guessed '0' or '1.' By comparing the relationship of the different accuracy metrics to each other, the alpha values, and the baseline metrics, we determined that the most optimal alpha value was 22.5. This is also the natural ratio of negative to positive cases, so the alpha value of 22.5 equally balances the two classes. Looking at the accuracy metrics, we can see a non monotonic increase for recall and a somewhat monotonic increase for precision and balanced accuracy. Additionally, we plot the independent relationship between precision and recall, as shown below.

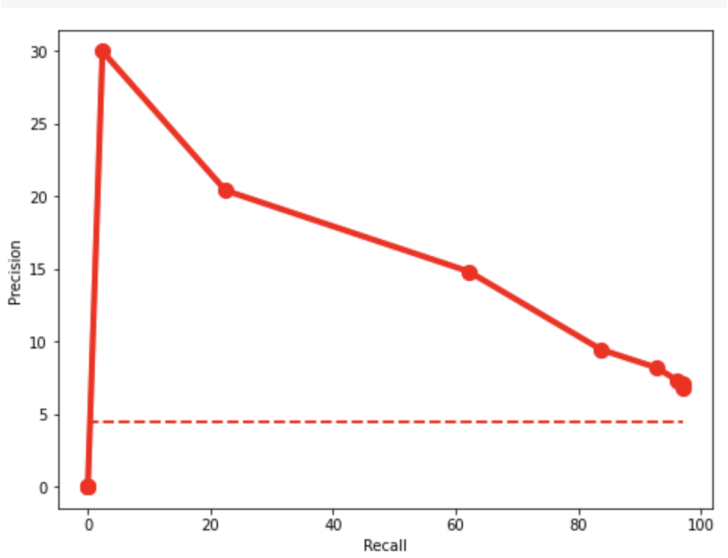

Figure 2: A precision vs recall graph displaying the tradeoff between accuracy scores for individual alpha values.

Finally, in an attempt to determine feature importance, we use scikit-learn's Random Classifier to determine the mean decrease in impurity (MDI), or the total decrease in node impurity averaged over the tree for a given feature. In doing so, we create the chart shown below for a given feature's mean decrease in impurity and how the importance compares in relation to each other.

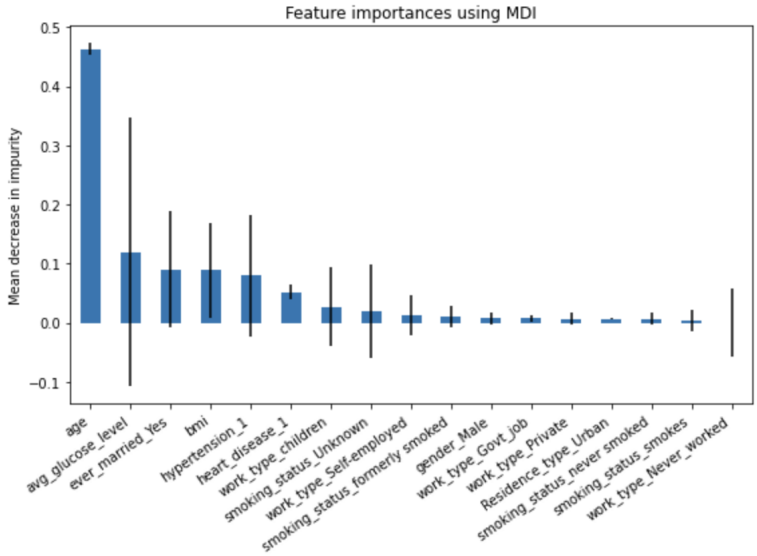

Figure 3: Bar graph displaying feature importance in descending order with confidence intervals using mean decrease in impurity.

Notably, the most important feature in stroke detection is age. Since there is an overlap in confidence intervals beyond the age feature, the subsequent most important features cannot be determined. However, age being the critical feature gives unique insight on which features

should be considered first when determining the probability of an individual having a stroke in the future.

6. Conclusions

As stroke becomes more prevalent in our society today, the need for preventative measures becomes even more crucial in order to save those who may be more susceptible to strokes and the adverse consequences that follow. Thus, by using a Random Forest Classifier in order to isolate the most important feature that should be taken into consideration, age, we can help improve stroke detection. Additionally, we utilized class weights in order to balance a severely imbalanced dataset.

In the future, we can pursue more state of the art results with larger datasets and more tuned models. However, these steps will require a good understanding of the weights, preprocessing of the features, and choice of evaluation metrics. For instance, we could compare different models. Additionally, finding a bigger dataset can be helpful in increasing the accuracy of the model. Currently, only a fraction of the data points are available to be used but increasing this number will give the models a larger number of data points to evaluate.

Acknowledgments

Inspirit AI Mentor MB had a large influence on this project, and was of great help for every stage of the research.

References

1. I. T. Akbasli, “Brain stroke prediction dataset,” Kaggle, 16-July-2022. [Online]. Available: https://www.kaggle.com/datasets/zzettrkalpakbal/full-filled-brain-stroke-dataset. [Accessed: 01-Nov-2022].
2. IEEE Xplore Full-text PDF: [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9264165. [Accessed: 01-Nov-2022].
3. S. Dev, H. Wang, C. S. Nwosu, N. Jain, B. Veeravalli, and D. John, “A predictive analytics approach for stroke prediction using machine learning and Neural Networks,” Healthcare Analytics, 15-Feb-2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772442522000090#b12. [Accessed: 01-Nov-2022].