Imbalanced Classification Techniques to Assess Stroke Risk

Matthew Hureau, Amr Ibrahim
Stony Brook University

INTRODUCTION

Stroke, also known as brain attack, occurs when blood supply to the brain is blocked or when a blood vessel in the brain bursts, damaging parts of the brain [1]. Strokes can lead to permanent brain damage, long-term disability, or even death. According to the CDC, stroke is a leading cause of death in the United States with around 800,000 strokes per year in the country, of which more than 140,000 die, and many survivors are faced with disability [2]. While the second leading cause of death in the world, about 80% of strokes are preventable [2]. We seek to analyze a dataset containing information on twelve variables for 5110 patients in order to build a predictive model to assess a new patient's risk of stroke, and to determine what variables have the greatest impact on one's likelihood of suffering one.

OBJECTIVES AND CONTRIBUTIONS

The main objective of this project was to build the best predictive model to assess stroke risk. A more general objective of potentially equal importance was for our group to gain experience dealing with health and healthcare-related data, as it's of interest to both of our graduate work. Many machine learning projects in this field happen to share a common obstacle for the data analyst, and that is dealing with the issue of imbalanced classification.

The term is most frequently used to describe a severely unequal distribution of classes in a dataset, particularly for the outcome variable one is trying to model. If the prevalence of a particular disease in a population is low, meaning the disease is not all that common, then the distribution of those who have and don't have the disease in the training set will very likely be severely imbalanced. Not many observations, meaning patients in our case, will have the disease, compared to the amount of patients who do not. A machine learning model fit using this imbalanced data will likely suffer from sensitivity issues, that being a low percentage of positive cases that the model correctly predicts as positive. The model has many negative case observations in the training data teaching it how to correctly predict that majority class, but a relatively small amount of observations of the minority class. We also seek to learn how to remedy this phenomenon with the appropriate techniques in order to make ourselves more well rounded data analysts.

While both team members had a part to play in each part of this analysis, we both had our own areas of focus. Amr came into this project with a more intensive background in Python and data visualization, while Matt had significantly more experience with R and machine learning. Each member was tasked with laying the groundwork in their respective areas, but also made sure to explain their code and intuition in detail to the other. We felt this was a good way to learn from one another while also maximizing efficiency.

TECHNIQUES AND TOOLS

The analysis was performed using both the Python and R programming languages. With most of the visualization being done in the former, and the machine learning in the latter.

We had remote bi-weekly meetings via zoom and wrote our Python code in a shared notebook via a website called Deepnote. We transferred our code from Deepnote to a Jupyter Notebook file as requested by our instructor before submitting our work. We wrote our R code in a markdown file in RStudio, and knit into a Microsoft Word document for submission.

This section of the report will be divided into three subsections. The first will describe the techniques and tools used for data cleaning, the second will do the same for data visualization, and the third for imbalanced classification techniques. The techniques and tools used for fitting the machine learning methods will be discussed in the results section.

(I) Data Cleaning

To our surprise there was not much data cleaning that needed to be done. The **bmi** predictor containing the body mass index of a patient was the sole variable containing any missing values. The 201 patients that were missing a BMI composed just under 4% of the total number of patients. The commonly accepted threshold for statisticians and data analysts to decide whether or not removing observations with missing variables outright is acceptable is 5%. We could have chosen to remove these observations but decided it would be beneficial to get hands-on experience with imputation. Instead of utilizing the MICE package in R to impute, we wanted to first visualize this continuous variable and see if a more basic method of imputation could work. When creating a histogram for this variable we noticed that its distribution seemed highly non-normal. We applied a Shapiro-Wilk Test in order to quantify this non-normality, which returned a p-value less than $2.2 \, x \, 10^{-16}$, indicating with extreme confidence that, in fact, the data did not come from a normal distribution. Upon learning this, we decided that it would not be smart to implement mean imputation, and instead imputed all of the missing values with the median BMI. Because the proportion of patients with missing values was small however, the method of imputation would likely have little to no influence on the outcome.

Among the 5110 observations only one had a non-binary gender. After discussing the potential risk-reward of doing so, we felt it would be easier to remove this one observation, and do the analysis using the available data for the remaining 5109 patients. Before moving on to visualization we decided to drop the variable containing the patient ID, as we felt intuitively that it played no role in determining the likelihood of a patient having a stroke.

(II) Data Visualization

Our ten predictor variables (once the patient ID is dropped) include three continuous variables, and seven categorical. We first visualized the distribution of the outcome variable, in this case whether or not the patient had a stroke. Only 4.87% of the 5109 patients suffered from a stroke, signifying that the data was heavily imbalanced (see fig. 1).

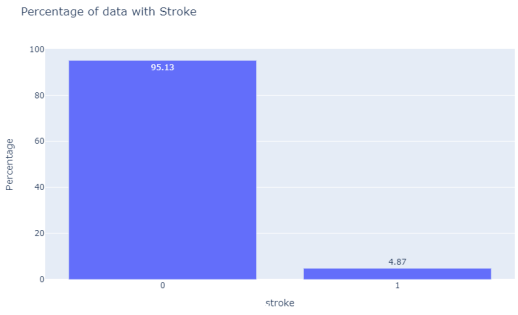

FIG 1: Distribution of Stroke Outcomes

We then visualized the predictor variables to assess which variables may have been correlated with the outcome. Age seemed to be the most

visually predictive among them, with the majority of stroke patients falling between the ages of 59 and 82 (see fig. 2). BMI and the average glucose level did not appear to have any major significance.

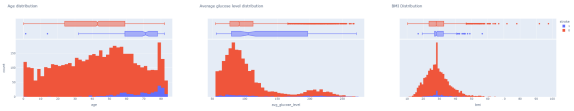

FIG 2: Distribution of Continuous Predictors

It was slightly more difficult to interpret the visualization of the categorical predictors, especially with those that were non-binary. **hypertension** and **heart_disease** were the health-related binary predictors. Stroke patients had increased hypertension and heart disease rates (see fig. 3). The increased hypertension percentage was larger compared to the increased heart disease percentage, suggesting that hypertension was the stronger predictor.

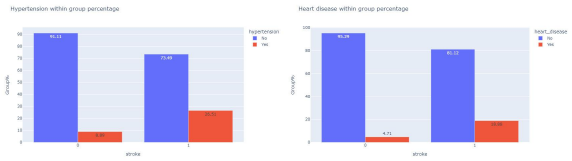

FIG. 3: Distribution of Health-Related Binary Categorical Predictors

gender, **ever_married**, and **Residence_type** were the three lifestyle-related binary predictors. Those variables were not as predictive as those that were health-related, where for most variables, there was no significant increase or decrease in percentage (see fig. 4). **ever_married** had an increased married percentage in stroke patients compared to non-stroke patients suggesting that **ever_married** is a stronger predictor.

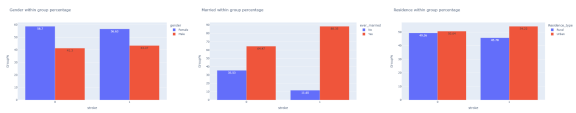

FIG. 4: Distribution of Lifestyle-Related Binary Categorical Predictors

Finally, the non-binary predictors were **smoking_staus** and **work_type**. These were more difficult to interpret, since most percentage changes from healthy to stroke were minimal. Those that were significant may have been affected by a different variable. Such phenomena is observed in **work_type** where stroke patients had an increased percentage of self-employed and decreased percentage of children which could possibly be a factor of age rather than work environment (see fig. 5). **smoking_status** contained a level called “unknown”, whose relative percentage decreased in stroke patients, signifying that formerly smoking patients is a strong predictor within **smoking_status**.

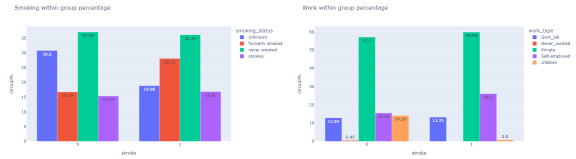

FIG. 5: Distribution of Non-Binary Categorical Predictors

(III) Imbalanced Classification

We implemented four core methods to deal with the imbalanced classification problem. The methods of oversampling, undersampling, a combination of over and under sampling, and synthetic data generation were each implemented on our training set after splitting the original data into 75% training and 25% testing.

Undersampling is the process of randomly sampling observations from the majority class until you have an equal amount of observations from each class. It can also be thought of as

randomly removing observations from the majority class until the classes are balanced. This method yields the smallest amount of total observations, thus a loss of information. Oversampling randomly makes copies of samples from the minority class until the number of observations in both classes is balanced. This method can often yield models that overfit the data, especially if done incorrectly. The general concepts of both oversampling and undersampling are demonstrated in the figure below. Combining both over and under sampling combines the two, randomly sampling from the majority class and randomly copying from the minority class. The final method we implemented was synthetic data generation. This method generates completely new observations based on the pre-existing training observations.

FIG. 6: Simple Demonstration of Undersampling and Oversampling

The implementation of these methods in Python was done using base functions. In R, the functions that were applied to the training set to implement each of the four methods were obtained via the **ROSE** package.

RESULTS

We considered at least eleven machine learning models that were implemented across the four types of imbalanced classification methods, resulting in a total of 45 potential models to choose from, each having their own sensitivity, specificity, and overall model accuracy. We implemented 10-fold cross validation for each of the models that required parameter tuning, such as Ridge and Lasso, Random Forest, and K-Nearest Neighbors. After we completed the data cleaning we split the data into 75% training and 25% testing. We then implemented the four imbalanced classification methods on the training set alone. We first decided to split **after** applying those techniques on the entire dataset, but we were left with overfitting issues exclusively with the methods implementing some amount of oversampling. Applying these four methods after splitting on exclusively the training set fixed this issue.

The computational complexity of fitting these models for the undersampled data was relatively simple, but changed drastically for the other three methods, which required at least some amount of oversampling techniques. Our undersampled training set had 182 observations of both outcome class 0 and 1 (372 observations total). The other three methods yielded significantly larger training sets. With the number of observations being in class 0 and 1 respectively, our oversampling training set had 3651, and 3649 observations, our over/under sampling combination training set had 1970, and 1862 observations, and our synthetic data training set had 1971, and 1862 observations. This caused a few of the machine learning models implemented on the larger training sets to take a significant amount of time to execute. In special cases such as the support vector machine with a radial or polynomial kernel, the code hadn't finished executing after approximately twenty minutes. Due to this, the support vector machine with a non-linear kernel was excluded from our consideration for the three techniques that required some form of oversampling. Applying a Neural Network with anything more than one neuron and one hidden layer did not converge after quite some time for these larger training sets, so for simplicity, each

of the neural network models contained one hidden layer and one neuron.

Interpretability can be lost with certain models, so we will only be discussing model interpretation for logistic regression. This model was the first we chose to fit, and the results of this model are shown below.

```
##
## Call:
## glm(formula = stroke ~ ., family = "binomial", data = training.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0310  -0.7888   0.1778   0.8243   2.5175
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -2.380772   0.967823  -2.460   0.0139 *
## genderMale                    -0.160190   0.265378  -0.604   0.5461
## age                            0.071396   0.010100   7.069 1.56e-12 ***
## hypertension                   0.242258   0.321646   0.753   0.4513
## heart_disease                 -0.084672   0.373546  -0.227   0.8207
## ever_marriedYes               -0.020911   0.417553  -0.050   0.9601
## work_typeGovt_job             -1.991576   1.049762  -1.897   0.0578 .
## work_typeNever_worked        -12.604695 882.743924  -0.014   0.9886
## work_typePrivate              -1.231746   1.013303  -1.216   0.2241
## work_typeSelf-employed        -2.000617   1.080368  -1.852   0.0641 .
## Residence_typeUrban            0.027337   0.258127   0.106   0.9157
## avg_glucose_level              0.005341   0.002503   2.134   0.0329 *
## bmi                           -0.027488   0.021680  -1.268   0.2048
## smoking_statusnever smoked    -0.004514   0.332441  -0.014   0.9892
## smoking_statussmokes           0.194115   0.401774   0.483   0.6290
## smoking_statusUnknown         -0.470086   0.388051  -1.211   0.2257
## ---
```

FIG. 7: Summary Table of the Logistic Regression Model

One should interpret the coefficients of this model as follows. A unit increase in age will result in an increase of 0.071396 in the log odds of having a stroke. For multi-factor categorical predictors, such as gender, the interpretation is slightly different, as it is measured against a baseline level. Looking at **genderMale**, going from female to male results in a decrease in the log odds of having a stroke by 0.16. It seems that with all other variables held constant, females are more likely to have a stroke than males.

Only two predictors were statistically significant at a significance level of 0.05, those being age and avg_glucose_level. In addition, three alternative Logistic Regression methods were fit to each training set. Stepwise selection, backward, and forward selection were each implemented and yielded the same results. For the undersampled data, the final model contained only age, avg_glucose_level, and work_type. Applying these three variable selection techniques significantly increased model sensitivity across each of the four training sets. These methods increased sensitivity by just over 6% for the undersampled training set, and at least 10% for the remaining methods. Penalized Logistic Regression methods Lasso, Ridge, and Elastic Net Regularization were also applied. These three methods significantly improved model sensitivity for each of the three oversampling methods, but did not have any change on the sensitivity of the undersampled model.

In addition to these seven logistic regression based methods, the Neural Network, Support Vector Machine, K-Nearest Neighbors, and Random Forest algorithms were also applied to each of the four training sets. As mentioned previously the Support Vector Machine with the radial basis kernel was only applied in addition to the linear kernel for the undersampled training set. Though using the radial basis kernel made only a slight change in model performance. Due to the large number of models fit, we decided to append the exhaustive performance results to the end of this report as a supplement. For this section instead, we chose to display what we deemed to be our best four models for each of the four imbalanced classification methods. These are displayed in the tables below.

UNDERSAMPLED:

	Model Accuracy	Sensitivity	Specificity
Stepwise	0.7565	0.7910	0.7546
Backward Selection	0.7565	0.7910	0.7546
Forward Selection	0.7565	0.7910	0.7546
Random Forest	0.7424	0.8209	0.7380

OVERSAMPLED:

	Model Accuracy	Sensitivity	Specificity
Forward Selection	0.7565	0.7910	0.7546
Support Vector Machine (Linear Kernel)	0.7447	0.7612	0.7438
Lasso	0.7659	0.7463	0.7669
Stepwise	0.7651	0.7463	0.7661

OVER/UNDER SAMPLED:

	Model Accuracy	Sensitivity	Specificity
Ridge	0.7682	0.7313	0.7703
Stepwise LR	0.7674	0.7313	0.7694
Lasso	0.7674	0.7313	0.7694
Backward Selection	0.7674	0.7313	0.7694

ROSE SYNTHETIC DATA GENERATION:

	Model Accuracy	Sensitivity	Specificity
K-Nearest Neighbors	0.7753	0.7463	0.7769
Support Vector Machine (Linear Kernel)	0.7619	0.7463	0.7628
Lasso	0.7729	0.7313	0.7752
Stepwise LR	0.7729	0.7313	0.7752

As a whole, the undersampling method yielded the best models. Each of the other three results in at least two models that performed very poorly. Most of the best models across each of the four methods were simply extensions of logistic regression. This is a good thing because of how simple it is to interpret the results of this algorithm relative to other methods. However, while the undersampling method worked best for our data, we would advise those working in a professional setting to take advantage of every ounce of time they are given to be exhaustive in their search for the best model. While there was clearly one method that worked best in our case, a different method may be optimal for the data they are working with.

CONCLUSION:

We seeked to build the best model to predict the likelihood of a patient having a stroke based on a finite number of risk factors, and to implement four different methods to deal with the problem of imbalanced classification, a phenomenon that is very common when dealing with predictive modeling for a particular disease. How old a patient is seems to be by far the most important variable in determining that risk, followed further behind by their average glucose level, and followed even further behind by what type of job they have. While the undersampling technique worked best with the data we chose to analyze, we recommend that the data analyst should be as exhaustive in their search for the optimal model as they can be, and should implement as many of these techniques as possible to do so.

References

[1] Centers for Disease Control and Prevention. (2021, August 2). About stroke. Centers for Disease Control and Prevention. Retrieved April 2, 2022, from https://www.cdc.gov/stroke/about.htm

[2] Centers for Disease Control and Prevention. (2017, September 6). Preventing stroke deaths. Centers for Disease Control and Prevention. Retrieved April 2, 2022, from https://www.cdc.gov/vitalsigns/stroke/index.html#:~:text=Almost%20800%2C000%20people%20have%20a,treatable%20risk%20factor%20for%20stroke.

FULL RESULTS:

UNDERSAMPLING:

	Model Accuracy	Sensitivity	Specificity
Logistic Regression	0.7612	0.7313	0.7628
Stepwise LR	0.7565	0.7910	0.7546
Backward Selection	0.7565	0.7910	0.7546
Forward Selection	0.7565	0.7910	0.7546
Neural Network	0.7619	0.7313	0.7636
Random Forest	0.7424	0.8209	0.7380
K-Nearest Neighbors	0.6938	0.7910	0.6884
SVM (Linear)	0.7463	0.7463	0.7463
SVM (Radial)	0.7478	0.7463	0.7479
Lasso	0.7518	0.7313	0.7529
Ridge	0.7572	0.7313	0.7587
Elastic Net	0.7416	0.7313	0.7422

OVERSAMPLING:

	Model Accuracy	Sensitivity	Specificity
Logistic Regression	0.8363	0.6418	0.8471
Stepwise LR	0.7651	0.7463	0.7661
Backward Selection	0.7651	0.7463	0.7661
Forward Selection	0.7565	0.7910	0.7546
Neural Network	0.7674	0.7463	0.7686
Random Forest	0.9405	0.0597	0.9895
K-Nearest Neighbors	0.9139	0.1940	0.9537
SVM (Linear)	0.7447	0.7612	0.7438
SVM (Radial)	-	-	-
Lasso	0.7659	0.7463	0.7669
Ridge	0.7572	0.7463	0.7579
Elastic Net	0.7643	0.7463	0.7653

OVER/UNDER SAMPLING:

	Model Accuracy	Sensitivity	Specificity
Logistic Regression	0.8371	0.5821	0.8512
Stepwise LR	0.7674	0.7313	0.7694
Backward Selection	0.7674	0.7313	0.7694
Forward Selection	0.7674	0.7313	0.7694
Neural Network	0.7682	0.7313	0.7703
Random Forest	0.9303	0.1791	0.9179
K-Nearest Neighbors	0.8786	0.3134	0.9099
SVM (Linear)	0.7510	0.7463	0.7512
SVM (Radial)	-	-	-
Lasso	0.7674	0.7313	0.7694
Ridge	0.7682	0.7313	0.7703
Elastic Net	0.7518	0.6866	0.7554

ROSE SYNTHETIC DATA GENERATION:

	Model Accuracy	Sensitivity	Specificity
Logistic Regression	0.8410	0.5373	0.8579
Stepwise LR	0.7729	0.7313	0.7752
Backward Selection	0.7729	0.7313	0.7752
Forward Selection	0.7729	0.7313	0.7752
Neural Network	0.7729	0.7313	0.7752
Random Forest	0.8003	0.5970	0.8116
K-Nearest Neighbors	0.7753	0.7463	0.7769
SVM (Linear)	0.7619	0.7463	0.7628
SVM (Radial)	-	-	-
Lasso	0.7729	0.7313	0.7752
Ridge	0.7706	0.7313	0.7727
Elastic Net	0.7533	0.7015	0.7562