CIS 520 Fall 2019, Final Project Report

Stroke Prediction: A Case Study in Class Imbalance

Team Members:

- Charles Marx; Email: chmarx@seas.upenn.edu
- Matthew Scharf; Email: mschar@seas.upenn.edu
- Sehyeok Park; Email: spark423@seas.upenn.edu

Abstract

We study the task of machine learning-assisted stroke prediction. The healthcare dataset we explore details demographic and medical information as well as a binary indicator for whether each individual experienced a stroke. The dataset studied in this project elicits challenges to accurate prediction common in stroke related data and medical data in general such as omission of data and stark imbalance between negative and positive cases. To deal with the issue of missing values, we explore different imputation methods in the preprocessing stage. After the initial preprocessing steps, we construct and apply the following models: logistic regression, AdaBoost, decision trees, neural network, support vector machines, k-nearest neighbors, and random forests. For each model, we explore asymmetric loss functions to account for the class imbalance present in the original dataset and the different costs of stroke diagnoses. We also explore different hyperparameters specific to each model to assess their impact on performance. We assess the performance of each model and class weighting using metrics such as the AUC, true positive rate, false positive rate, precision, recall, and accuracy. The results indicate that the more complex models employed in the project do not noticeably outperform their simpler counterparts in classification power. Given these findings, we conclude that methods such as logistic regression and decision trees are our models of choice for this task due to their interpretability and ease of adoption.

1 Motivation

Strokes are the fifth leading cause of death among Americans and impose a socioeconomic burden costing the United States $34 billion each year [1]. Given the serious implications on public health, improving the accuracy of stroke prediction is an increasingly critical task in medicine. The motivation of this project is to assess the performance of various machine learning methods in forecasting the occurrence of a stroke for a patient given key social demographic and medical predictors. With advancements in medical recording improving the availability of patient data and stroke diagnoses necessitating the consideration of a variety of potential predictors, machine learning is a promising approach to tackling the task of stroke prediction. The overlap between common obstacles to predictive medical diagnoses and recurring themes in applied machine learning exemplified by data omission and data imbalance further highlights the utility of machine learning in addressing the issue at hand.

2 Related Work

Strokes are a significant health risk worldwide, and as a result the evaluation of individual stroke risk has been of growing interest. Previously, researchers have attempted to employ machine learning to predict the occurrence [5, 3] and mortality [2] of strokes. Existing work largely focuses on comparing the effectiveness of standard machine learning models for stroke prediction, focusing primarily on the predictive performance of each model. As a precursor to developing and assessing model performance, existing literature makes effort to address machine learning challenges inherent to the task of making predictions based on medical records and surveys. To overcome omissions in clinical data, Khosla et al. [5] explore various methods of data imputation ranging from mean imputation to regression based imputation, both of which will be explored in this paper. Cheon et al. [2] explore issues stemming from class imbalance in stroke mortality prediction resulting from the small portion of fatal strokes. Similarly, we will address the low ratio of stroke occurrence within the dataset during the preprocessing and training stages and also explore performance metrics suitable for interpreting outputs from imbalanced data.

In addition to a comparison of existing methods for stroke prediction, we intend to contribute (1) a discussion of model interpretability and (2) an evaluation of model performance under an asymmetric loss function, both of which to the best of our knowledge have little presence in the current literature.

3 Data Set

We study the the Healthcare Dataset Stroke Data.[1] The dataset includes 43,300 patient observations each labeled with a binary indicator representing stroke occurrence. Of the 43,300 instances, 783 (1.9%) have a positive label for stroke. In addition to the stroke labels, the dataset includes ten predictors, which are as follows: gender, age, hypertension, heart disease, marriage history, occupation type, residence type, average glucose level, body mass index, and smoking status. For the gender predictor, 59.14% of the observed individuals are female, 40.84% are male, and 0.03% are labeled as "other". The minimum, maximum, mean, and standard deviation of the reported age are 0.08, 82.00, 42.22, and 22.52 respectively.

The dataset also includes information regarding medical conditions. The data includes binary indicators for both hypertension and heart disease. The percentages of the observations that have positive labels for hypertension and heart disease are each 9.36% and 4.75%. The minimum, maximum, mean, and the standard deviation of the reported average glucose level are 55.00, 291.05, 104.48, and 43.11 respectively. For the last two predictors, body mass index and smoking status, there are missing data. The dataset is missing body mass index data for 3.37% of the observations. The minimum, maximum, mean, and the standard deviation of the BMI feature are 10.10, 97.6, 28.61, and 7.77 respectively. With respect to smoking status, 30.63% of the observations are missing. When present, smoking status takes values among "currently smoking", "formerly smoked", and "never smoked". The aforementioned levels correspond to 15.12%, 17.26%, and 36.99% of the observations respectively.

[1]The dataset is available at `https://www.kaggle.com/asaumya/healthcare-dataset-stroke-data`.

3.1 Data Visualization

For the task of visualization we use PCA. To display the trends in the raw data we derived the first two principle components and plot them according to whether they correspond to a reported case of stoke occurrence. The resulting plot substantiates the aforementioned class imbalance between reported strokes and non-strokes and indicates there being no apparent trend or separation between the two distributions. For the purposes of visualization, we plot all points labeled stroke=1

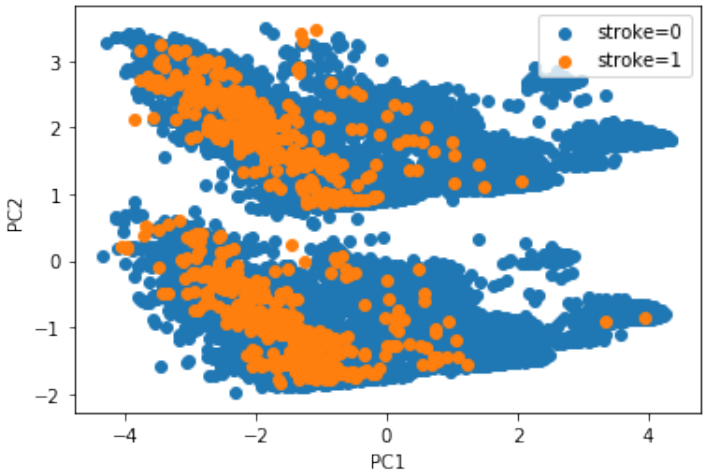

Figure 1: PCA Visualization of the data colored by stroke occurrence.

4 Problem Formulation

Our objective is to identify a binary classification model for stroke prediction. First, we will preprocess the data and impute the 'bmi' and 'smoking status' features which are missing 3.4% and 30.6% of the instances in the training data respectively. We will train a variety of models which given data, X, provide a binary classification, y, which minimizes an asymmetric loss function. We will need to find the correct asymmetric loss function which represents the real-world costs of stroke diagnoses and takes into account the presence of class imbalance. After we have fit our models, we will evaluate the trade-off for different models between classification performance and interpretability. Finally, we will provide insight into the structure of the data through visualization.

5 Methods

5.1 Preliminaries

We consider a dataset $D = (x_i, y_i)_{i=1}^n$ where $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip}) \in \mathbb{R}^p$ and $y_i \in \{0, 1\}$. Denote the matrix of features by $X = (x_i)_{i=1}^n$ and the vector of binary labels by $y = (y_i)_{i=1}^n$. For the stroke dataset, the number of instances is $n = 43,300$ and the number of features after preprocessing is $p = 20$.

5.2 Preprocessing

Missing Value Imputation. The dataset has two features with missing data: BMI (numeric) and SMOKING STATUS (categorical). For the numeric feature BMI, we use regression imputation to estimate missing values. After imputing BMI, we impute the SMOKING STATUS feature using logistic regression. We opted to regress missing features as opposed to mean value or most common value imputation in order to leverage information provided by the other features.

Normalization. To prepare the data for analysis, we one-hot encode all categorical features and explore two strategies for normalizing numeric features. First, we one-hot encode all categorical features that have no missing values (GENDER, MARRIAGE HISTORY, OCCUPATION TYPE, and RESIDENCE TYPE). We then impute the missing data for the BMI and SMOKING STATUS features using the imputation methods mentioned in the previous section. After filling in the missing values, we one-hot encode the imputed feature SMOKING STATUS.

To normalize the data, we first explore *standardizing* each feature to have mean=0 and variance=1 using the transformation

$$x_{ij} = \frac{x_{ij} - \overline{x}_j}{\sigma_j}$$

where $\overline{x}_j = \frac{1}{n}\sum_{k=1}^{n} x_{kj}$ is the sample mean for feature j and $\sigma_j = \sqrt{\frac{\sum_{k=1}^{n}(x_{kj} - \overline{x}_j)^2}{n-1}}$ is the sample standard deviation for feature j. Second, we separately consider *min-max scaling* each feature to have a minimum value of 0 and a maximum value of 1 via the transformation

$$x_{ij} = \frac{x_{ij} - \min_k x_{kj}}{\max_k x_{kj}}$$

While there are advantages and disadvantages to each normalization strategy [4], we find that normalizing using the mean=0, var=1 procedure yields the best results for logistic regression on our data. Thus, we use standardization as our normalization procedure in all of our experiments.

5.3 Learning Algorithms

5.3.1 Models

We will explore eight prediction models, which are as follows: deep neural network, random forest, logistic regression, Adaboost classifier, k-nearest neighbors, support vector machine, rule list, and decision tree. We implement each algorithm using the `scikit-learn` package [6].

5.3.2 Class Weighting

Due to the severe class imbalance (1.8% positive instances) and the higher cost of false negatives (erroneously predicting no stroke) relative to false positives (predicting a stroke which will not occur), we incorporate class weighting into our loss function. For a loss function $\mathcal{L}$, vector of labels y, and vector of predictions $\widehat{y}$ and we say that

$$\mathcal{L}_\alpha(y, \widehat{y}) = \frac{\alpha}{n} \sum_{y_i=1} \mathcal{L}(y_i, \widehat{y}_i) + \frac{(1-\alpha)}{n} \sum_{y_i=0} \mathcal{L}(y_i, \widehat{y}_i)$$

where α is a parameter which tunes the importance of correctly predicting the positive class relative to the importance of correctly predicting the negative class. The special case of $\alpha = 0.5$ balances the importance of predicting each class correctly, and for values of $\alpha \geq 0.5$ we increase the importance of predicting the positive class correctly relative to the negative class.

5.3.3 Exhaustive Hyperparameter Searches

For each learning algorithm we consider, we define a space of possible model settings using a free hyperparameter grid. We exhaustively search all combinations of hyperparameters for each learning algorithm, and choose the best performing set of hyperparameters using 3-fold cross-validation (using more folds was computationally infeasible given the number of models and available computing resources). After selecting the best performing hyperparameters using cross-validation we retrain a model on the full training set using the optimal hyperparameters and report the performance of this model on the test set.

6 Experiments and Results

6.1 Comparison of Preprocessing Methods

We must impute the categorical SMOKING STATUS feature and the continuous BMI feature in addition to normalizing all features. We compare all possible combinations of normalization procedures (standardization, min-max scaling), continuous imputation methods (zero, mean, linear), and categorical imputation methods (mode, logistic) described in section 5.2. In Figure 2, we show the optimal test set performance achieved by a logistic regression model fit using an exhaustive CV hyperparameter search for each preprocessing pipeline.

In the remainder of our experiments we train all models using standardization, zero imputation for BMI, and logistic regression imputation for SMOKING STATUS since these choices

	meanvar						minmax					
	linear		mean		zero		linear		mean		zero	
	mode	logistic	mode	logistic	mode	logistic	mode	logistic	mode	logistic	mode	logistic
accuracy	0.948732	0.948461	0.978299	0.948393	0.947172	0.947308	0.948461	0.948461	0.905941	0.978503	0.946358	0.946426
auc	0.616805	0.616667	0.525503	0.614626	0.634069	0.640157	0.614661	0.612654	0.671289	0.521593	0.639674	0.639709
f1_score	0.150562	0.149888	0.080460	0.147816	0.163265	0.168984	0.147982	0.146067	0.131497	0.070381	0.166491	0.166667
precision	0.103876	0.103236	0.135922	0.101852	0.110787	0.114493	0.102009	0.100775	0.077663	0.125000	0.112216	0.112376
recall	0.273469	0.273469	0.057143	0.269388	0.310204	0.322449	0.269388	0.265306	0.428571	0.048980	0.322449	0.322449

Figure 2: Metrics for Logistic Regression (with CV hyperparemter search) trained with different combinations of preprocessing techniques. The column levels correspond to the normalization technique, the numerical imputation method, and the categorical imputation method respectively.

resulted in the best performance. We treat logistic regression as our baseline throughout the following experiments.

6.2 Model Performance

We next explore how the performance varies according to learning algorithm, class weighting, hyperparameter choice, and performance metric. For each learning algorithm we perform an exhaustive search over class weights and hyperparameter settings to optimize the AUC. The chosen hyperparameters for each model are shown in Table 3.

	Chosen Hyperparameters
Logistic Regression	C=0.5, class_weight={1:5}, l1_ratio=0.7
AdaBoost	learning_rate=1.0, n_estimators=20
Decision Tree	class_weight={1:10}, criterion=gini, max_depth=3
Neural Network	hidden_layer_sizes=(10,10)
k-Nearest Neighbors	n_neighbors=50
Support Vector Machine	class_weight={1:50}, kernel=linear
Random Forest	max_depth=5, n_estimators=200

Figure 3: Hyperparameters chosen by a CV grid search for each learning algorithm.

Test Set Metrics	AUC	TPR	FPR	precision	recall	accuracy
Logistic Regression	0.763	0.763	0.238	0.762	0.763	0.762
AdaBoost	0.773	0.804	0.258	0.757	0.804	0.743
Decision Tree	0.751	0.906	0.405	0.691	0.906	0.600
Neural Network	0.763	0.784	0.258	0.752	0.784	0.743
k-Nearest Neighbors	0.729	0.837	0.379	0.688	0.837	0.625
Support Vector Machine	0.769	0.788	0.250	0.759	0.788	0.750
Random Forest	0.764	0.800	0.271	0.747	0.800	0.730

Table 1: Test set model performance according to each metric. Thresholds were chosen for each model to maximize precision while maintaining a recall of at least 0.75 on the training set.

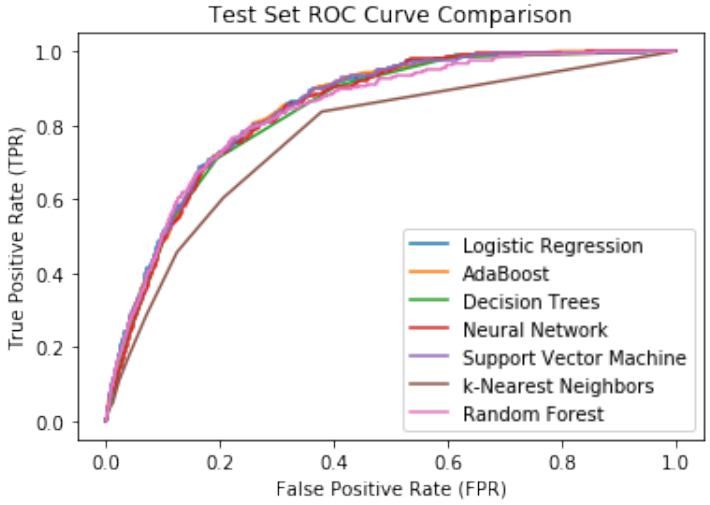

Figure 4: ROC Curves

Using the hyperparameter setttings specified in Table 3 we retrain each model on the full training set. Test set metrics for these models are reported in Table 1, and in the ROC curve in Figure 4. To tune the trade-off between precision and recall for each model, we choose a threshold which maximizes precision with the constraint of maintaining a recall of at least 0.75. We see that with the exception of the k-Nearest Neighbors model which performs worse, all models perform similarly. This may be because the signal present in the data is mostly present in one or two features in simple patterns, and all models are similarly capable of utilizing this information (e.g., older people are more likely to have strokes than younger people). Since no model shows a significant improvement over the logistic regression baseline, we advocate for using simple and interpretable models such as logistic regression and small decision trees. We include a visualization of a decision tree classifier in Figure 5. We see that the decision tree mostly relies on two features, AGE and SEX, supporting the hypothesis that a small set of features contain most of the signal.

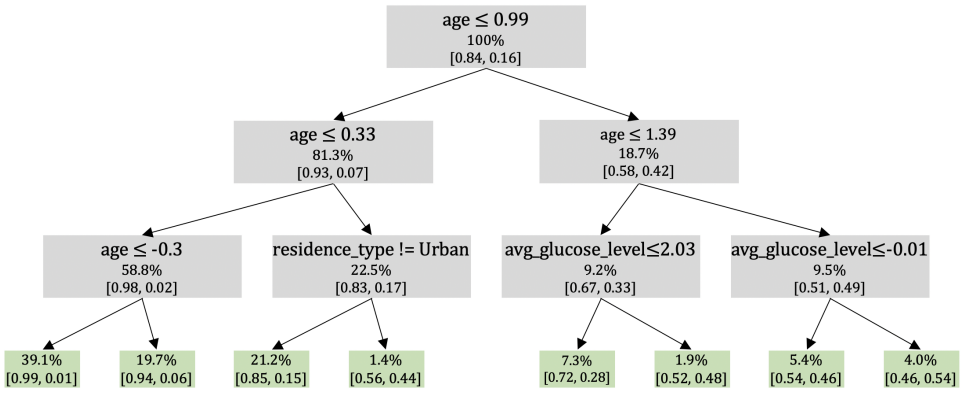

Figure 5: A decision tree classifier for predicting stroke occurrence. Numerical features are presented in their normalized form. The numbers below each node in the tree represent the percent of training data and distribution of labels in each partition. The classifier achieves an AUC of 0.751 on the test set.

7 Conclusion and Discussion

7.1 Model Interpretability

Interpretability of a model is a measure of how well someone using the model can understand its decision-making process. This is especially important for diagnostic models because local interpretability provides explainability for patient treatment and global interpretability provides insight for clinical researchers to better understand what is being diagnosed.

One possible downside of interpretable models (e.g. a decision tree) is that they are typically limited in their complexity and so may not have the same classification power as a less interpretable model (e.g. a neural network). However, in this case, the decision tree performs comparably with less interpretable alternatives and so, due to its interpretable nature, is our model of choice.

Acknowledgments

We would like to thank Saumya Agarwal for sharing the stroke dataset.

References

[1] Stroke statistics. *Centers for Disease Control and Prevention*, Sep 2017. url: https://www.cdc.gov/stroke/facts.htm.

[2] Songhee Cheon, Jungyoon Kim, and Jihye Lim. The use of deep learning to predict stroke patient mortality. *International journal of environmental research and public health*, 16(11):1876, 2019.

[3] Cemil Colak, Esra Karaman, and M Gokhan Turtay. Application of knowledge discovery process on the prediction of stroke. *Computer methods and programs in biomedicine*, 119(3):181–185, 2015.

[4] Anil Jain, Karthik Nandakumar, and Arun Ross. Score normalization in multimodal biometric systems. *Pattern recognition*, 38(12):2270–2285, 2005.

[5] Aditya Khosla, Yu Cao, Cliff Chiung-Yu Lin, Hsu-Kuang Chiu, Junling Hu, and Honglak Lee. An integrated machine learning approach to stroke prediction. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 183–192. ACM, 2010.

[6] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.